

UN MODELO PARA LA GENERACIÓN DE EXPLICACIONES EN SISTEMAS DE RECOMENDACIÓN

ALBERTO MERINO DE LA FUENTE

MÁSTER EN INVESTIGACIÓN EN INFORMÁTICA, FACULTAD DE INFORMÁTICA,
UNIVERSIDAD COMPLUTENSE DE MADRID



Trabajo Fin de Máster en Sistemas Inteligentes

Junio 2012
Calificación: 8.0

Director:
Pedro Antonio González Calero

Colaborador:
Héctor Gómez Gauchía

Autorización de Difusión

ALBERTO MERINO DE LA FUENTE

20 de Junio de 2012

El abajo firmante, matriculado en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: “UN MODELO PARA LA GENERACIÓN DE EXPLICACIONES EN SISTEMAS DE RECOMENDACIÓN”, realizado durante el curso académico 2011-2012 bajo la dirección de Pedro Antonio González Calero y la codirección de Héctor Gómez Gauchía en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Índice

Autorización de Difusión	i
Resumen en castellano	v
Palabras clave.....	v
Abstract	vii
Keywords	vii
Introducción	1
Capítulo 1 – Estado del Arte	5
1.1 Explicaciones en sistemas expertos	5
1.1.1 Explicaciones en sistemas de reglas.....	6
1.1.2 Explicaciones en razonamiento basado en casos	11
1.2 Tipos de sistemas de recomendación.....	22
1.2.1 Recomendadores colaborativos.....	22
1.2.2 Recomendadores basados en contenido	24
1.2.3 Recomendadores basados en conocimiento	26
1.2.4 Recomendadores híbridos	30
1.3 Explicaciones en sistemas de recomendación	34
1.3.1 Objetivos de la recomendación	35
1.3.2 Explicaciones colaborativas.....	38
1.3.3 Explicaciones basadas en contenido	41
1.3.4 Explicaciones basadas en conocimiento	54
1.3.4 Comparativa de los modelos estudiados	57
Capítulo 2 – Modelos de conocimiento propuestos	63
2.1 Modelo de conocimiento de un sistema de recomendación	64
2.2 Modelo de conocimiento para el Ingeniero de conocimiento	71
2.3 Modelo de conocimiento de la explicación	73
Capítulo 3 – Aspectos de diseño	75
Capítulo 4 – Modelo de explicación propuesto	79
4.1 Introducción	79
4.2 HotSpot Explanations: Un modelo para generar explicaciones.....	80

Capítulo 5 – Implementación.....	85
5.1 Caso de estudio Moodvie.....	85
5.2 Explicaciones para Usuarios Finales	94
5.3 Explicaciones para Ingeniero del Conocimiento	106
Capítulo 6 – Conclusiones y trabajo futuro	125
6.1 Conclusiones.....	125
6.2 Trabajo futuro	125
Referencias.....	127

Resumen en castellano

Los sistemas de recomendación como sistemas de filtrado que presentan los ítems en los que un usuario estará interesado, por lo tanto, un aspecto muy importante es la generación de explicaciones que expliquen el porqué de una recomendación. Las explicaciones son muy usadas en las tareas de comunicación y de razonamiento entre las personas, por lo tanto, incorporar esta faceta en los sistemas de recomendación es una mejora muy importante.

En este trabajo presentamos HotSpot Explanation, un modelo que permite la generación de explicaciones para los sistemas de recomendación que tengamos implementados. El modelo reconstruye el proceso de recomendación y permite la generación de una explicación al usuario de porqué un determinado ítem es adecuado para él usando la reconstrucción de ese proceso. También permite la generación de explicaciones para los ingenieros del conocimiento de forma que puedan verificar los datos y el proceso de razonamiento del sistema.

Palabras clave

Sistemas de recomendación, explicaciones.

Abstract

Recommender systems as filtering systems that presents items in which a user will be interested, therefore, a very important aspect is the generation of explanations that explain why of a recommendation. Explanations are widely used in commutations and reasoning task between people, therefore, incorporate this aspect in recommender systems is a significant improvement.

In this paper we present HotSpot Explanations, a model that allows generation of explanations for recommender systems that we have implemented. The model reconstructs recommendation process and allows generation of explanations to the user. Explanations are a *Why?* recommended item is suitable for the user using the reconstructed process. It also allows generations of explanations for knowledge engineers, so that they can check data and the reasoning process of the system.

Keywords

Recommender systems, explanations.

Introducción

Los sistemas de recomendación se están convirtiendo en los últimos años en unas herramientas fundamentales y bastante comunes, sobre todo en los sistemas web debido a la capacidad que tienen para personalizar los contenidos a los usuarios. Ejemplos de este éxito lo podemos ver en sistemas que los han implementado como *Amazon*, *Facebook*, *Youtube*, *Last.Fm*, etc.

Otro sistema de recomendación y es por el que surge este proyecto es Moodvie (Garrido, Peña, Salmerón, 2011) se trata de un recomendador de películas, donde se nos plantea el reto de cómo poder mejorar este sistema y la principal idea que se nos ocurre es la de incorporar explicaciones al sistema que por un lado permitan mostrar al usuario porque debería ver una película y por otra a los desarrolladores del sistema les permitiera verificar el razonamiento seguido por el sistema debido a que muchas veces no presentaba los resultados esperados.

También observamos que esto es una característica bastante común en la mayoría de los sistemas que hay implementados actualmente, muy pocos presentan explicaciones y creemos que este es un aspecto bastante importante, ya que la explicación es un aspecto explotado en la comunicaciones humanas y sobre todo cuando hablamos en un contexto de recomendaciones.

Este trabajo de fin de máster propone un modelo “HotSpot Explanations” que nos permita generar explicaciones en sistemas de recomendación que ya estén implementados, este modelo se basa en reconstruir la línea de razonamiento seguida por el sistema de recomendación y generar por un lado explicaciones para los usuarios finales de porque deberían usar un determinado ítem usando esa línea de razonamiento reconstruida. Por otro lado al reconstruir la línea de razonamiento también podremos presentar explicaciones para el ingeniero del conocimiento de cómo se ha producido el proceso de recomendación de forma que le permita verificar el sistema. Otro aspecto positivo es que este modelo nos permite desacoplar el proceso de recomendación del de generación de la explicación.

El objetivo principal de este trabajo es llevar a cabo una investigación sobre los diferentes mecanismos que existen para generar explicaciones en sistemas de recomendación, este objetivo a su vez se divide a su vez en los siguientes objetivos específicos:

- Realizar un estado del arte, hemos realizado una revisión de los siguientes campos de investigación: Explicaciones en sistemas expertos, tipos de sistemas de recomendación y explicaciones en sistemas de recomendación.
- Hacer un análisis y estudio detallado de las explicaciones y del dominio de conocimiento de los sistemas de recomendación, presentando una serie de modelos de ese conocimiento. Con todo esto presentar un modelo que nos permita generar explicaciones en los sistemas de recomendación.
- Aplicar este modelo a un sistema de recomendación existente, en nuestro caso Moodvie, mostrando como ha sido la implementación.

El contenido de esta memoria está estructurado de la siguiente manera:

- En el capítulo 1 se ha llevado a cabo el estado del arte, sobre los diferentes temas tratados en el proyecto. Primero una revisión sobre las explicaciones en los sistemas expertos, tanto en los sistemas de reglas como en razonamiento basado en casos. Después se hará una revisión de los diferentes tipos de sistemas de recomendación que hay haciendo especial hincapié en cómo funcionan. Para finalizar una revisión de las explicaciones en los sistemas de recomendación, presentado cuáles son sus objetivos y para cada uno de los tipos de sistemas de recomendación anteriores una revisión detallada de artículos que presentan explicaciones de esa naturaleza.
- En el capítulo 2 hemos querido hacer un análisis del dominio de conocimiento de los sistemas de recomendación y presentar una serie de modelos de conocimiento genéricos tanto para usuarios finales como para los ingenieros del conocimiento, ya que estos serán de vital importancia para poder generar las explicaciones en el modelo propuesto.
- En el capítulo 3 hemos querido hacer un análisis de los diferentes aspectos de diseño que tenemos que tener en cuenta a la hora de diseñar las explicaciones.
- A continuación en el capítulo 4 presentamos un modelo para la generación de explicaciones en sistemas de recomendación, mostrando como sería el proceso de diseño y después la arquitectura del sistema implementado.
- Por último, en el capítulo 5 mostramos un ejemplo de la implementación del modelo anterior en un ejemplo real, en este caso Moodvie.

- Para concluir, en capítulo 6 se muestran las conclusiones finales del trabajo y cuáles son las futuras líneas de trabajo futuro.

Capítulo 1 – Estado del Arte

En este capítulo hacemos una revisión actual del campo de investigación en el que está inmerso este proyecto. Hemos querido dividirlo en tres sub secciones, en la primera haremos una pequeña revisión de las explicaciones en los sistemas expertos más tradicionales, haciendo especial hincapié en los sistemas de reglas ya que estos fueron los primeros en abordar el tema de las explicaciones y sus trabajos han tenido gran influencia en las investigaciones actuales. El otro sistema experto elegido ha sido el razonamiento basado en casos, esta técnica es también muy interesante de analizar desde el punto de explicaciones ya que muchos de los sistemas de recomendación actuales utilizan esta técnica como estrategia de filtrado. En la segunda sub sección presentamos una revisión del estado actual de los sistemas de recomendación, haciendo énfasis en los diferentes tipos de estrategias más representativas. Para finalizar, y la más importante ya que es la que atañe al objetivo principal de la investigación, una descripción de los modelos más representativos en el área de las explicaciones en los sistemas de recomendación.

1.1 Explicaciones en sistemas expertos

Las explicaciones en sistemas inteligentes no son nuevas, surgen precisamente en el área de investigación de los sistemas expertos en especial de los sistemas de reglas (Clancey, 1983), (Swartout, 1983), (Chandrasekaran, Tanner y Josephson, 1989). Las investigaciones en sistemas expertos se han centrado sobre todo en las explicaciones que se pueden generar y como se han implementado en sistemas reales.

En nuestro trabajo hemos hecho una revisión de las explicaciones en sistemas de reglas debido a la importancia histórica que ha tenido y del razonamiento basado en casos por su relevancia en el área de los sistemas de recomendación. Tenemos revisiones más completas sobre explicaciones en sistemas de reglas (Lacave y Diez, 2004) y razonamiento basado en casos en (Doyle, Tsymbal, Cunningham, 2003) y (Sormo, Cassens y Aamodt, 2005).

Otro sistemas experto importante son las redes bayesianas pero no lo hemos incluido en nuestro trabajo, aunque si el lector está interesado en explicaciones sobre este método puede encontrar una completa revisión en (Lacave y Diez, 2002).

1.1.1 Explicaciones en sistemas de reglas

Los primeros sistemas con explicaciones surgen con el sistema de reglas MYCIN que se trataba un sistema para detectar y recomendar tratamientos en enfermedades infecciosas (Clancey, 1983). En este sistema el usuario podía hacerle preguntas de cómo había llegado a una determinada conclusión y el sistema le respondía con la traza del razonamiento que había usado a modo de explicación.

Todas estas explicaciones se realizaban de un modo textual, utilizando un lenguaje natural y números para expresar el grado de certeza, esto ofrecería al usuario un grado de transparencia en cómo el sistema había llegado a sus conclusiones. El usuario también puede elegir una explicación de porqué el sistema le había hecho una pregunta determinada.

Pronto se encontró que esta capacidad tenía defectos importantes, en primer lugar, el sistema no estaba seguro si los usuarios entienden el proceso de encadenamiento de reglas, también en ocasiones es complicado conseguir un argumento coherente cuando el número de reglas que encadena es alto y por último, era insuficiente para responder a muchas de las peticiones de explicaciones de los usuarios. Por ejemplo, la estrategia de resolución de problemas de un sistema experto basado en reglas se define implícitamente en el sistema, pero no fue codificado explícitamente de forma que sea accesible o fácil de explicar a un usuario final.

NEOMYCIN (Clancey, 1983) amplió las capacidades de MYCIN en este aspecto, fue desarrollado con el objetivo de explicar los resultados de MYCIN. Contenía un conocimiento explícito casual, representado por meta-reglas y una taxonomía de las enfermedades, de forma que el objetivo no es solo explicar la cadena de razonamientos, sino también la estrategia de razonamiento. Por lo tanto la finalidad es la comprensión tanto del modelo como del razonamiento. Las explicaciones son presentadas mediante texto, en lenguaje natural.

Otra extensión de MYCIN fue XPLAIN (Swartout, 1983), la idea principal está basada en usar un programa automático para crear el sistema experto y la explicación de su razonamiento por el refinamiento de objetivos abstractos. XPLAIN tenía los siguientes elementos:

- El dominio de modelo, que contenía los hechos descriptivos del dominio y su representación mediante una red.
- Los principios del dominio del razonamiento, definidos por varios métodos y heurísticas.

- El programa automático para la generación del programa experto con el conocimiento proporcionado en los dos tipos de dominios anteriores.
- La estructura necesaria para generar la explicación, representada mediante un árbol, que se desarrolla mediante la construcción del sistema experto.
- El traductor para expresar en lenguaje natural las explicaciones, toma como entrada el árbol generado anteriormente. Tiene dos elementos, un generador de frases, para construir frases directamente de la base de conocimiento y un generador de respuesta, que determina que partes de la estructura deben ser incluidos en la explicación y el nivel de detalle, teniendo en cuenta las necesidades de los usuarios y el estado del sistema.

El objetivo de este método es la comprensión del modelo y del razonamiento, mediante explicaciones generadas que analizan las principales líneas de razonamiento y presentadas en un lenguaje natural. Por otra parte, el nivel de detalle se modifica durante la ejecución del sistema, dependiendo de las necesidades del usuario.

El objetivo de estas extensiones tempranas fue por lo general ampliar las capacidades de la explicación añadiendo el tipo de conocimiento requerido por el usuario. Estas explicaciones se pueden dividir en cuatro tipos según el contenido que tengan (Chandrasekaran, Tanner y Josephson, 1989; Gregor y Benbasat, 1999):

- Traza del razonamiento. Produce una explicación de la traza del proceso de razonamiento utilizado por el sistema para encontrar una solución. Ejemplos, son las explicaciones de tipo *Why?* y *How?* usadas por el sistema MYCIN.

La primera permite al usuario preguntar por qué una información es solicitada por MYCIN, en la Figura 1.1 vemos un ejemplo de este tipo de explicación, vemos que cuando el sistema pregunta si J.Smith es un huésped inmunodeprimido (pregunta 23), el usuario responde al sistema que porqué hace esa pregunta en ese contexto (*Why?*), después el sistema le responde para que usara esa información. Para explicarlo, MYCIN muestra la regla heurística que considera en particular la 343 y explica como la información le será útil, después volverá a la pregunta de partida (23) donde como vemos el usuario puede seguir preguntando por qué y el sistema le sigue respondiendo pero en ese caso porque es interesante saber la conclusión de la regla 343, porque es interesante saber si es inmunosupresor y

continua el proceso, con lo cual podemos seguir una traza del proceso de razonamiento del sistema.

Figura 1.1 Explicación de tipo *Why?* en MYCIN (Clancey, 1983)

```

23) Is J.Smith a compromised host?
** WHY

[i.e. WHY is it important to determine whether J.Smith is a compromised
host?]

[3.0] This will aid in determining whether J.Smith is immunosuppressed.
If
  [3.1] J.Smith is not a compromised host
  then
    it is definite ( 1.0 ) that J.Smith is not immunosuppressed
  [RULE343]

[back to question 23...]
** WHY

[i.e. WHY is it important to determine whether J.Smith is immunosuppressed?]

[4.0]... in order to determine whether J.Smith has received corticosteroids.
If
  [4.1] J.Smith is not immunosuppressed
  then
    it is definite ( 1.0 ) that J.Smith has not received corticosteroids.
  [RULE395]

[back to question 23...]
** WHY

[i.e. WHY is it important to determine whether J.Smith has received
corticosteroids?] .

[5.0]... in order to determine the organisms causing the infection.

It has already been established that
  [ 5.1 ] the Infection which requires therapy is meningitis, and
  [ 5.2 ] only circumstantial evidence is available for this case, and
  [ 5.3 ] the type of meningitis is bacterial,

Therefore, if
  [ 5.4 ] J.Smith is receiving corticosteroids

then
  there is evidence that the organisms causing the Infection are:
    e.coli (.4), klebsiella-pneumoniae (.2), or pseudomonas-aeruginosa (.1)
  [RULE543]

[back to question 23...]
** NO

Consultation excerpt showing explanation capability
(User input in BOLD ITALICS; all other text is printed by MYCIN.)

```

El segundo método de explicación sería el basado en la pregunta *How?*, en este caso el usuario quiere saber como el sistema ha llegado a una conclusión.

- Justificación. Proporcionar una justificación para un paso del razonamiento vinculándolo con el conocimiento profundo del cual se derivó. XPLAIN fue el primer sistema que las introdujo. Dentro de este tipo de explicaciones hay a su vez cuatro forma de obtenerlas:

- Siendo informado directamente. El conocimiento solo puede justificarse apelando a la autoridad, ejemplo, “mira el libro X, pag 90”.
- Generalizando a partir de ejemplos. Por ejemplo, “La última vez que un paciente tenía estos síntomas, resultó ser cáncer”.
- Por inferencia explicativa. Cuando por ejemplo, un sistema que contiene una regla que (teniendo ciertas síntomas) un microorganismo infeccioso debe ser la hipótesis. Esto se justifica por el conocimiento de que, si la enfermedad estuvo presente, esto explicaría los síntomas.
- Mediante derivación de una comprensión más profunda del dominio. Cuando, por ejemplo, "Las reducciones de impuestos en general, fomentan el ahorro, estimulan la inversión, y aumentan la producción (lo que disminuye los precios, las exportaciones aumentan, hace a los productos nacionales atractivos, y reduce el déficit comercial)".

Si la base de conocimientos contiene los datos relativos a que los recortes de impuestos reducen el déficit comercial, tal razonamiento no está obligado a concluir que una reducción de impuestos es adecuado. Sin embargo, el sistema debe tener esos conocimientos a disposición para que pueda ser utilizada adecuadamente para su explicación.

- Estrategia o control. Explicación de la estrategia de razonamiento del sistema o para controlar el comportamiento del razonador. Ejemplo NEOMYCIM.

Estas explicaciones puede explicar algunas preguntas tipo “¿Por qué no...?” así como “¿Por qué...?”. Lo que se necesita es la capacidad de abstraer y encajar partes de la estrategia de control a la situación, por ejemplo la siguiente situación:

Pregunta: ¿Por qué no está sugiriendo un aumento de las tarifas como una manera de disminuir el déficit comercial?

Explicación: Porque el plan implica costes políticos. Mi estrategia es considerar los planes políticamente más fácil primero.

- Terminológica. Definir y explicar los términos y conceptos del dominio, por ejemplo:

Pregunta: ¿Qué es la sensibilidad a los fármacos?

Explicación: La sensibilidad a los fármacos es una desviación observable que causa algo peligroso que es causado por la droga.

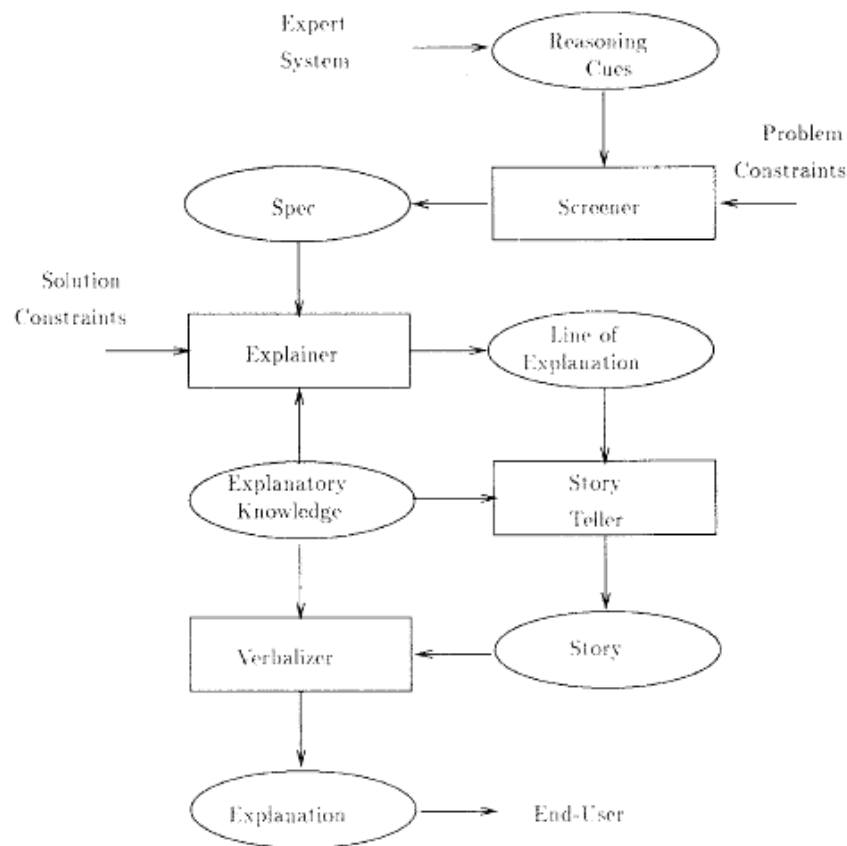
El posterior análisis de sistemas expertos sugiere que muchas de estas explicaciones fracasaron debido a que o eran incomprensibles para los usuarios o no tenían en cuenta los objetivos de los usuarios en la demanda de las explicaciones (Majchrzak y Gasser, 1991). En respuesta a esto, la investigación adicional fue en cómo podrían las explicaciones generarse dinámicamente para adaptarse a las necesidades de los usuarios y sus objetivos.

(Wick y Thompson, 1992) sugieren otro método para realizar las explicaciones, su propuesta habla de la reconstrucción de la misma, donde consideran el proceso de explicación y razonamiento como dos tareas independientes con el fin de que las explicaciones se centren en la solución, su argumento es que las personas en sus explicaciones no describen todos los pasos para llegar a la solución. Por lo tanto, consideran que los métodos para la generación de explicaciones deben tomar como entrada el proceso de razonamiento realizado por el sistema para obtener la solución, teniendo en cuenta el dominio de la explicación. Además sugieren que las principales metas de las explicaciones sean:

- Verificación. Este es el objetivo del ingeniero del conocimiento, la verificación de que el sistema funciona como debería.
- Duplicación. Aquí se trata de ayudar al experto del dominio a examinar el conocimiento del sistema, el sistema no solo debe exponer el conocimiento, sino que debe ayudar al usuario a aprender los métodos y conocimiento usados en el proceso de resolución de problemas.
- Ratificación. Este es el objetivo de incrementar la confianza del usuario en la conclusión del sistema.

Todas estas ideas se ponen en práctica en el prototipo REX Figura 1.2, la idea del sistema es la siguiente: el sistema experto produce una serie de pista sobre el proceso de razonamiento que son filtradas para determinar que partes del razonamiento se pasaran al modulo de generación de explicaciones, teniendo en cuenta las restricciones del problema. Este modulo usa el dominio de explicación para obtener los conceptos de la explicación que justifiquen las pistas de razonamiento y luego otro modulo que las traduce a un lenguaje natural.

Figura 1.2 REX modelo para generar explicaciones en sistemas expertos (Wick y Thompson, 1992)



Como hemos visto a lo largo de esta sección la principal característica de estos métodos es que el sistema no tiene la capacidad de poder actualizar su conocimiento sobre el usuario durante la ejecución. Esto implica que el nivel de detalle de las explicaciones es fijo. En algunos de los más recientes el usuario puede variar el nivel de detalle pero en general sobre un conjunto pequeño de valores. La interacción entre el sistema y el usuario se realiza a través de preguntas o preguntas predefinidas.

1.1.2 Explicaciones en razonamiento basado en casos

Los sistemas de reglas y el razonamiento basado en casos presentan una similitud en el contexto, ambos se usan para proporcionar consejos a las personas sobre decisiones, debido a esto la tipología de las explicaciones en razonamiento basado en casos y sistemas de reglas serán muy parecidos. A partir de ahora nos referiremos a razonamiento basado en casos por sus siglas en ingles CBR.

En CBR la unidad básica de conocimiento es el “caso” al igual que hacíamos con los sistemas basados en reglas que usaba las reglas para generar explicaciones, en este caso usaremos los casos para producir las mismas. Según (Sormo, Cassens y Aamodt, 2005) la metodología de CBR parece muy transparente, el concepto de usar casos similares para resolver el problema actual es fácil de entender. Hay investigaciones que demuestran que mostrar los casos junto a su solución mejora la confianza de los usuarios en comparación a si solo se mostrara la solución (Cunningham, Doyle, Loughrey, 2003).

Además mostrar el caso recuperado al usuario es un tipo de comunicación del conocimiento que permite al usuario hacer su propio juicio de la similitud que existe en la vieja situación respecto a la nueva.

En ambos casos dependen de la habilidad del usuario para entender que es el caso y para confirmar la evaluación de similitud. La dificultad para el usuario en comparar casos se incrementa a medida que la estructura de los casos se vuelve más compleja y las medidas de similitud más complicadas, también se incrementa con el uso de técnicas de adaptación más complejas como cuando no se pueden recuperar casos que no sean parecidos pero que se ajustan al proceso de adaptación.

Con lo expuesto anteriormente parece que el caso no es la única fuente de conocimiento en un sistema CBR y además en muchas situaciones no nos proporcionan la información suficiente, debido a esto (Richter, 1995) identifica cuales son los contenedores de conocimiento en un sistema CBR y que se encuentran resumidos en la Figura 1.3 a continuación describimos cada uno de estos contenedores de conocimiento y como nos puede ser de utilidad para las explicaciones según el trabajo de (Roth-Berghofer, 2004):

- La base de casos, contiene los problemas concretos o prototipos que han sido resueltos previamente por el sistema. A menudo se suele decir que el caso sirve como una explicación en sí misma, esto es verdad en escenarios donde los casos tienen contenidos tanto descripciones del problema como de la solución, estos son conocidos como casos tipo regla. Pero existen otros casos conocidos como tipo restricción en los cuales, cual es la descripción del problema y cual es la solución solo se conoce en tiempo de ejecución, haciendo que simplemente presentar el caso como explicación sea bastante cuestionable, debido a esto se hace imprescindible otro tipo de estrategias de explicación.

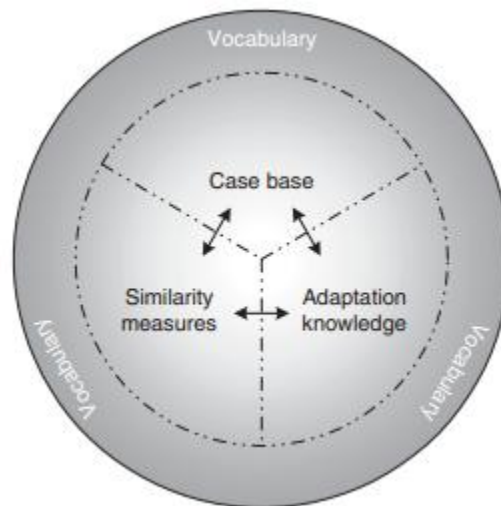
Por lo tanto el hecho de que existan diferentes tipos de casos tiene un impacto importante en que explicaciones se pueden construir. Encontrar el caso óptimo que resuelva el problema del usuario puede que no haga posible responder explícitamente a porqué ha sido seleccionado o como. Con lo cual ese tipo de cuestiones deben de ser proporcionadas mediante los otros tres tipos de contenedores de conocimiento.

- El vocabulario, es la base de los contenedores de conocimiento definiendo los términos y estructuras del conocimiento. Algunos sistemas CBR emplean un enfoque basado en objetos o marcos para poder capturar el conocimiento del dominio en un modelo de dominio. Esto conceptos capturados ya se pueden representar mediante relaciones *es-un*, *tiene-parte* y junto las técnicas de herencia y descomposición no va permitir responder a cuestiones sobre ¿Cuál es la estructura del caso?, ¿Qué partes del caso son específicas al caso clase y cuales son heredaras?, ¿Dónde está situado caso clase situado en el modelo de dominio? Además cada objeto o marco de un concepto va poder contener valores de una determinada clase limitada por ciertas restricciones. Esto permite explicaciones que respondan a cuestiones ¿Qué valores son permitidos en este slot? Con lo cual contra mayor tengamos elaborado el modelo de conocimiento, mas poder explicativo tendrá nuestro sistema.
- La medida de similitud, contiene el conocimiento sobre como comprar casos y calcular un ranking de casos similares respecto a un nuevo problema. Las medidas de similitud pueden ser divididas a su vez en dos tipo:
 - Locales, estas contienen el dominio del conocimiento y lo que hacen es comparar un atributo de un caso, por ejemplo, la similitud de colores de coches. Por lo tanto esta medida nos proporciona la información que puede ser usada para explicar cómo los valores de un atributo están relacionados con otros. También puede explicar relaciones entre porciones de vocabulario.
 - Función de fusión, contiene la utilidad del conocimiento, por ejemplo, la importancia del atributo color del coche contra el fabricante del mismo.

Esta medida nos proporciona la información necesaria para explicar la importancia o relevancia de un atributo.

- La adaptación del conocimiento, permite al razonador cambiar la solución de un caso anterior para ajustarse mejor al nuevo problema. Este contenedor requiere el conocimiento más profundo del dominio para estar disponible y la mayor representación formal para ser aplicable. Por lo tanto, si este conocimiento está disponible mucho más se podrá explicar sobre el sistema CBR.

Figura 1.3 Contenedores de conocimiento en un sistema CBR (Roth-Berghofer, 2004)



En la Tabla 1.1 se resumen brevemente a qué tipo de explicación cada contenedor de explicación contribuye principalmente.

Tabla 1.1 Contenedores de conocimiento y su contribución a las explicaciones (Roth-Berghofer, 2004)

Knowledge container	contributes to
Vocabulary	conceptual explanations, why-explanations, how-explanations, and purpose explanations
Similarity measures	why-explanations, how-explanations, purpose explanations, and cognitive explanations
Adaptation knowledge	why-explanations, how-explanations, and cognitive explanations
Case base	(provides context for explanations)

En donde:

- *Why?, How?, Purpose-explanations*, corresponden con las explicaciones de traza de razonamiento, justificación y estrategia que vimos en la sección de explicaciones de sistemas expertos.
- *Conceptual explanations*, corresponderían a las explicaciones terminológicas que vimos en la sección de los sistemas expertos, explicarían las relaciones semánticas.
- *Cognitive explanations*, son muy parecido a las de tipo de *Why?*, quieren explicar el comportamiento del sistema inteligente, en su conjunto.

En (Sormo, Cassens y Aamodt, 2005) tenemos una revisión bastante completa de explicaciones en sistemas CBR, este trabajo nos muestra una serie de sistemas clasificados por la técnica que usan para realizar la explicación:

- Mostrando casos similares, la forma más común de explicación en los sistemas CBR es mostrar el caso más similar. Esta técnica ha sido usada por muchos sistemas como por ejemplo CARES (Ong *et al.*, 1997) cuyo objetivo es predecir la recurrencia de cáncer de colon. En la Figura 1.4 vemos un ejemplo de explicación basada en mostrar el caso junto al caso similar.

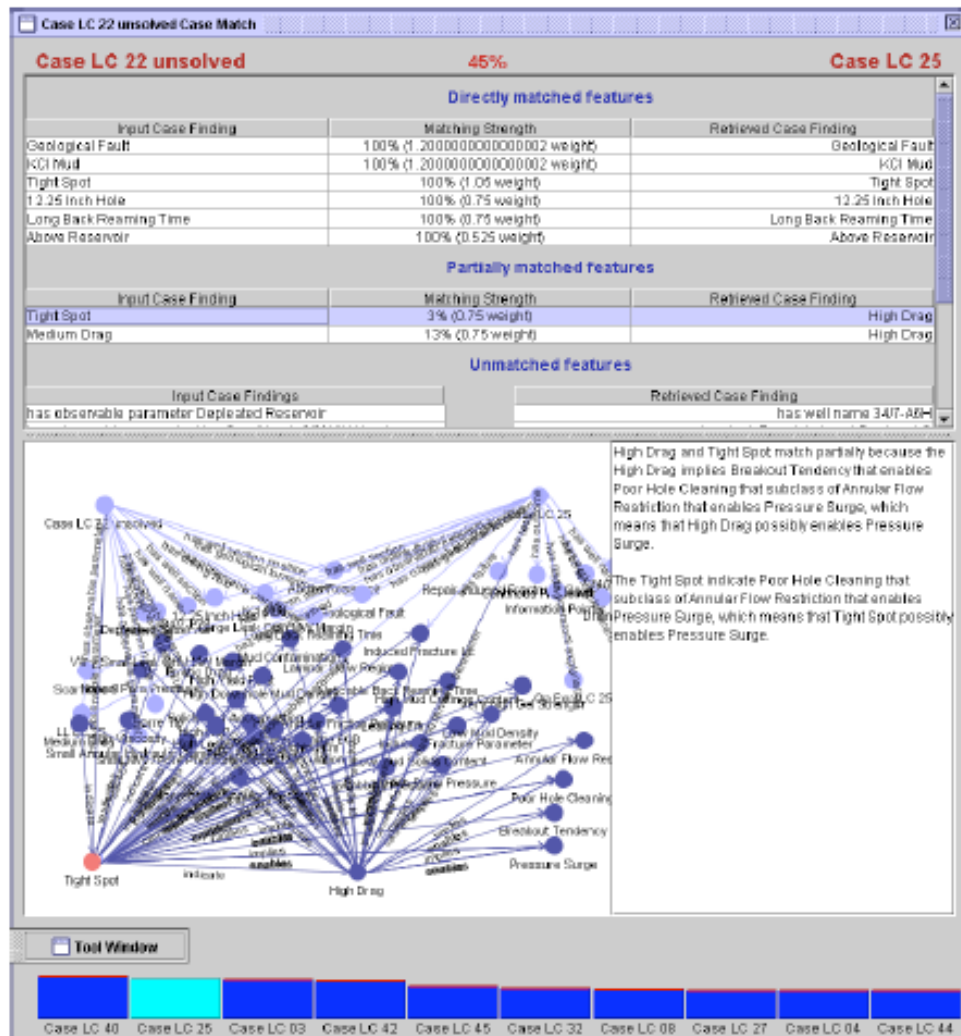
Figura 1.4 Ejemplo de explicación en CARES (Ong *et al.*, 1997)

Input Case	Stored Case	Similarity
duration of loss	C	C
duration of sym	0	0
ethnic group	C	C
extent of surgery	1	1
fib to adj struc	1	1
inv to adj struc	2	1
investigatin type	1	1
karnofsky perf sta folup	(100, 100, 100)	(80, 80, 80, 80)
length of tumour	4	1
liver folup	(False, False, False, F)	(False, False, False, F)
liver outcome	False	False
loss in kilogram	0	0
loss of appetite	False	False
loss of weight	False	False
loss of weight folup	(False, False, False, F)	(False, False, False, F)
loss of weight outcome	True	True
lymphatic invade	True	True
mass folup	(False, False, False, F)	(False, False, False, F)
mass outcome	True	False
mucous per rectum	False	False
number nodes inv	7	5
number of nodes exam	13	14
per rectum folup	(False, True, False, F)	(False, False, False, F)
per rectum outcome	True	False
perineural invade	False	False
proximal margin	11	4
radio anasto leak	False	False
rectal mass	False	False
recurrence detected folup	True	(False, False, False, F)
recurrence outcome	True	True
residual tumour folup	(False, False, True, F)	(False, False, False, F)

Aunque como vimos en los contenedores de conocimiento el uso del caso más cercano no puede servir ni de justificación del objetivo o de la transparencia. Una opción sería la de introducir diferentes capas de explicación en el proceso del CBR. El caso podría servir como un tipo de explicación de primer nivel, con niveles más detallados de explicación para cada característica del caso.

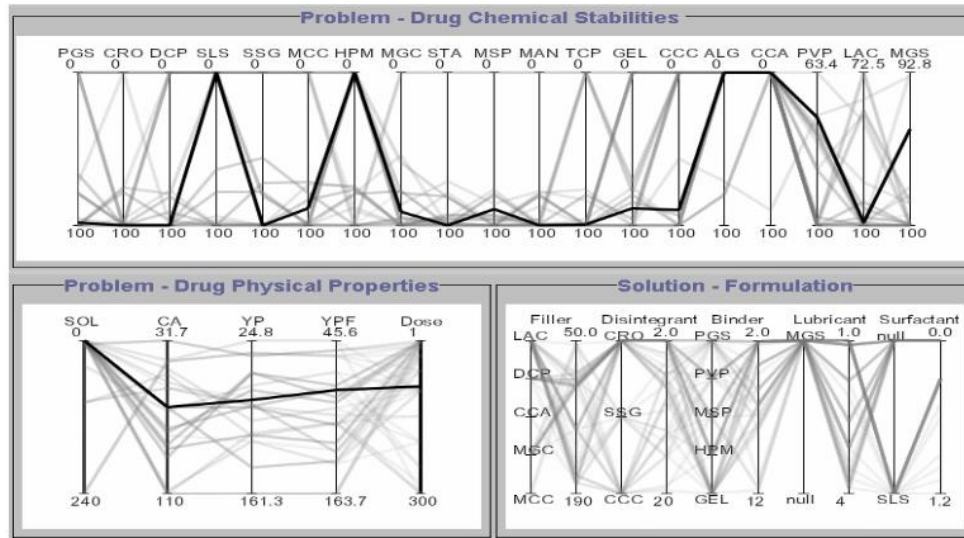
Un ejemplo de esta ultima idea sería el sistema CREEK (Aamodt, 2004) ver Figura 1.5 el usuario puede pedir explicaciones a nivel de atributos y la generación de la misma depende de la medida de similitud, un ejemplo seria cuando explica la similitud de los de los atributos en un rango de valores, el rango de esos atributos se muestra al usuario para que pueda ver más fácilmente lo similares que son en el contexto de casos conocidos.

Figura 2.5 Ejemplo de explicación en CREEK (Aamodt, 2004)



- Visualización, este método puede hacer que sea más simple para un usuario ver si una solución es correcta. Un ejemplo de esta técnica la vemos en el sistema FormuCaseViz (Massi, Craw y Wiratunga, 2004) ver Figura 1.6, es una herramienta que se usa para la formulación de tabletas farmacéuticas, en este sistema se visualiza como un número de casos difiere en una serie de atributos y como este conduce a las predicciones.

Figura 1.6 Ejemplo de explicación en FormuCaseViz (Massie *et.al*, 2004)



Este método se hace dibujando un grafico en dos dimensiones, donde cada atributo es representado mediante una línea vertical y sus valores se representan a lo largo de esa línea, un caso se representa como una línea horizontal que cruza las líneas de cada atributo en los puntos que representan el valor de caso para ese atributo. Esto permite a los usuarios unas simples comparaciones a simple vista.

- Traza de razonamiento, este método es factible en sistemas que producen explicaciones como parte del proceso de razonamiento. Un ejemplo de este método lo encontrado en el sistema de recomendación Top Cases (McSherry, 2005), este sistema sigue una estrategia en la cual va seleccionando preguntas que hacerle al usuario basándose potencialmente en las cuestiones que fortalecen el encuentro con las mejores hipótesis de un caso. Esta estrategia se le explica al usuario mostrando como una respuesta a una pregunta podría afectar a la recomendación. En la figura 1.7 tenemos un ejemplo de esta situación.

Figura 1.7 Ejemplo de explicación de traza de razonamiento en Top Cases (McSherry, 2005)

Top Case: Please enter your query
User: type = wandering, month = aug
Top Case: The target case is:
Case 510: 609, aug, tyrol, 5, 7, wandering, flat, car (0.28)
Other competing cases (972) include:
Case 702: 1698, aug, giant_mountains, 3, 14, wandering, star2, car (0.28)
Case 574: 529, aug, alps, 4, 7, wandering, flat, car (0.28)
What is the preferred location?
User: **why**
Top Case: Because if location = tyrol this will increase the similarity of Case 510 from 0.28 to 0.44 and eliminate 866 cases, including Case 574

En la imagen vemos que si el usuario quiere saber porque es relevante un lugar en este recomendador, el sistema le responde que si elige la opción Tirol haría aumentar la similitud del caso 510 y eliminaría 866 casos.

- Contrastar evidencias, que el objetivo de un sistema sea el de la transparencia exige que el sistema no trate de ocultar pruebas contradictorias sobre su recomendación. En CBR esto se puede lograr mostrando el caso similar pero que no son de la clase de la solución propuesta al usuario. Existen sistemas que siguen este enfoque como ProCon-2 (McSherry, 2004) un sistema de clasificación y diagnostico que permiten identificar que atributos de los casos apoyan la descripción del problema propuesta por el usuario y que atributos se oponen a ella. En la Figura 1.8 Tenemos un ejemplo del sistema.

Figura 1.8 Ejemplo de explicación en ProCon-2 (McSherry, 2004)

ProCon-2: Please describe the target problem.

User: weight = 79, duration = 90, sex = male, meal = full, units = 10.1

ProCon-2: The 3 most similar cases are:

Case 13: 79, 240, male, full, 9.6, over-limit (0.97)
Case 26: 73, 120, male, full, 9, not-over-limit (0.96)
Case 56: 79, 120, male, full, 7.2, not-over-limit (0.96)

The predicted outcome is: over-limit.

User: **why**

ProCon-2: Features of the target problem that SUPPORT the prediction:

units = 10.1

Features of the target problem that OPPOSE the prediction:

weight = 79
duration = 90
sex = male
meal = full

The most similar case is:

Case 13: 79, 240, male, full, 9.6, over-limit

The outcome in this case was over-limit in spite of the opposing evidence that:

weight = 79
sex = male
meal = full

En la figura nos muestra un ejemplo de su funcionamiento, en este ejemplo se le pide al usuario que introduzca la descripción del problema y el sistema le responderá con los casos más similares y con la solución de si ha sobrepaso el límite de alcohol en sangre. El usuario pide una explicación y el sistema le responde primero con las características que apoyan como las que no la solución, después para el caso más similar si cumple la característica de que sobrepasa el limite señala que lo cumple a pesar de que hay tres evidencias que en otros casos no se han cumplido.

- Simplificar la estrategia de resolución de un problema, la comunidad conversacional CBR ha desarrollado métodos que son específicos con el objetivo de una explicación relevante. Un ejemplo de esto es Strategist (McSherry, 2001) se trata de un sistema conversacional de diagnostico donde el usuario puede entrar en un dialogo con el sistema en donde este le va haciendo preguntas, el objetivo del mismo será proporcionar una buena explicación para cada pregunta de porqué es importante que el usuario las conteste.

Por ejemplo, en un dominio de fallos informáticos, en donde el usuario ha reportado al sistema que esta encendido y enchufado pero no funciona, la relevancia de la pregunta que hace el sistema “¿Puedes oír el ruido del ordenador?” ver Figura 1.9. Se podría explicar, en el contexto de otras pruebas, diciendo a los usuarios “Porque si el ventilador no se oye, esto podría confirmar que el cable de alimentación esta estropeado como posible causa” ver Figura 1.10.

Figura 1.9 Ejemplo de pregunta en Strategist (McSherry, 2001)

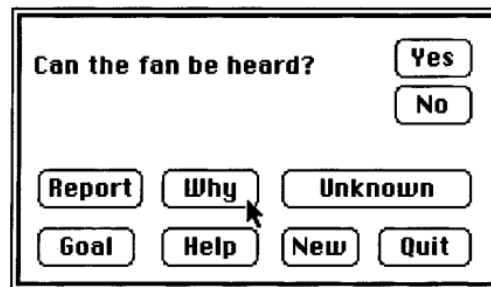
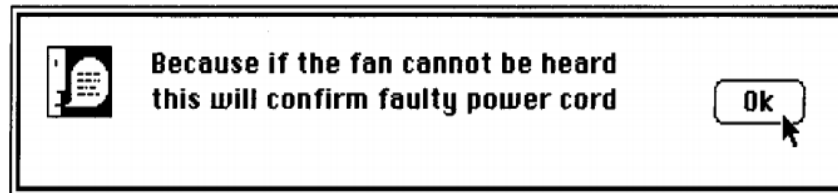


Figura 1.10 Ejemplo de explicación en Strategist (McSherry, 2001)



- El objetivo del aprendizaje parece tener una preferencia por los métodos de conocimiento intensivo, pero la generalización también se puede hacer de manera perezosa mediante una serie de algoritmos de aprendizaje automático. Los sistemas CBR Strategist y ProCon-2 de (McSherry, 2001; McSherry, 2004) son ejemplos de sistemas que utilizan la inducción al presentar una solución al usuario, pero lo hacen de una manera perezosa.

Como vimos en CBR Strategist todos los casos que tenia “No se puede escuchar el ventilador” tienen la misma solución “Cable de alimentación defectuoso” y puede informar al usuario de que esta solución es suficiente para confirmar la solución.

El sistema CBR Strategist puede ser bastante eficaz en la formación de los usuarios en la habilidad de detectar fallos en el ordenador. Una limitación de esto es que el sistema no puede introducir conceptos de alto nivel o que se refieran a como se usan los conceptos generales en ambientes externos al sistema.

1.2 Tipos de sistemas de recomendación

Un elemento clave para poder generar explicaciones en los sistemas de recomendación es entender cómo funcionan los mecanismos de recomendación asociados, para ello a lo largo de esta sub sección haremos un estudio de cuáles son las técnicas de recomendación más comunes basándonos en el libro de (Jannach, 2011), haciendo un análisis de cómo funcionan y para cada una de ellas presentaremos una serie de ejemplos de sistemas implementados que nos sirvan de referencia.

1.2.1 Recomendadores colaborativos

La idea básica de estos sistemas es que si los usuarios compartieron los mismos intereses en el pasado, por ejemplo, si vieron o compraron los mismos libros, también tendrán gustos parecidos en el futuro. Por lo tanto, por ejemplo, si un usuario A y un usuario B tienen un historial de compras que coincide fuertemente y un usuario A ha comprado recientemente un libro que todavía no ha visto B, lo lógico sería proponer este libro a B.

Los métodos colaborativos no explotan o requieren ningún conocimiento sobre los objetos, continuando con el ejemplo de la librería, el sistema de recomendación no necesita conocer de qué trata un libro, su género o quien los escribió. La ventaja obvia de estos sistemas es que estos datos no han de ser introducidos en el sistema o mantenerse. Por el contrario, usar estas características para proponer libros que son similares a los que le gustaba en el pasado puede ser más eficaz.

En cuanto al proceso de recomendación la técnica más común usada por estos sistemas es la basada en usuarios, que utiliza correlaciones entre un usuario **u** y otros usuarios, con el fin de decidir hasta que punto estos deberían influir en las recomendaciones para **u**, por ejemplo, en la Figura 1.11 queremos averiguar que puntuación daría Alice al ítem 5, primero tenemos que averiguar como de similares son cada uno de los usuarios a Alice en función de las puntuaciones

de los cuatro primeros ítems. Según (Tintarev, 2010) El algoritmo de recomendación puede ser dividido en los siguientes pasos:

1. Establecer la similitud de otros usuarios al del usuario u , una técnica muy usada es el coeficiente de correlación de Pearson, principalmente compara las puntuaciones que han dado a ítems para establecer como de similares son.
2. Utilizar esa similitud para que haga de peso sobre las puntuaciones de esos usuarios para el ítem recomendado i en forma de una suma ponderada, básicamente contra mas similares sea un usuario mayor será su peso en la recomendación.
3. Opcionalmente aplica filtros o pesos adicionales.

Figura 1.11 Ejemplo de puntuaciones sobre una serie de ítems (Jannach, 2011)

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Para nuestro ejemplo usando el paso 1 se determinan como de similares son cada usuario a Alice, en este caso vemos que el User2 y User3 son más influyentes, sus puntuaciones pasadas son más parecidas, cuando realicemos el segundo paso al ser más parecidos tendrán un peso más alto cuando realicen la predicción del ítem5.

Uno de los problemas de esta técnica es la necesidad de tener que analizar un gran número de vecinos, usuarios que tengan gustos parecidos, hace que con esta técnica sea prácticamente imposible hacer predicciones en tiempo real.

Para resolver este problema una técnica también muy usada en sitios de comercio electrónico es usar recomendaciones basadas en los ítems, esta técnica es más apta ya que permite un preprocesado offline de los ítems permitiendo que se puedan realizar recomendaciones en tiempo real. Estos algoritmos de recomendación pueden dividirse en los siguientes pasos (Tintarev, 2010):

1. Establecer cuantos ítems son similares al ítem **i**, lo normal es usar medidas como las del coseno.
2. Utilizar esta similitud con los pesos de la puntuación media (sobre todos los usuarios) para estos ítems con el fin de predecir la puntuación para el ítem recomendado **i** en forma de una suma ponderada.
3. Aplicar filtros o pesos adicionales.

Para el caso de la Figura 1.11 veríamos que ítems son mas similares entre si, en este caso, vemos que las puntuaciones que los usuarios dan sobre los ítems 1 y 5 son muy parecidas, con lo cual cuando calculemos la predicción de la puntuación para el ítem 5 le daremos un valor alto al peso del ítem 1.

En cuanto sistemas implementados que utilicen este filtrado para recomendar, el primer ejemplo que veremos es *MovieLens* (Herlocker *et.al*, 1999; Sarwar *et.al*, 2001), se trata de un recomendador de películas que usa un enfoque basado en los ítems, con lo cual el usuario inicialmente vota una serie de películas, entre un rango de uno (muy mala) a cinco (muy buena) después el sistema hará un preprocesado offline de los datos y encontrara los ítems que más se le parezcan en función de las puntuaciones que ha hecho la comunidad.

Otro trabajo interesante es el realizado por (Linden, Smith y York, 2003) en donde nos presenta como funciona el algoritmo de filtrado colaborativo basado en ítems usado por *Amazon*. Otro filtro colaborativo es el recomendador *Ringo* (Shardanand y Maes, 1995) es un recomendador de música y para finalizar el sistema de recomendación *Tivo* (Kamal y Wijnand, 2004) que se utiliza para recomendar programas de televisión.

1.2.2 Recomendadores basados en contenido

Del anterior tipo de recomendaciones vimos que a parte de las calificaciones que aporta el usuario no conocemos nada más sobre los ítems que recomendamos. Esto tiene una parte buena y es que no necesitamos mantener descripciones de los ítems pero también un aspecto negativo y es que no podemos recomendar productos a un usuario basándonos en sus características y en sus preferencias.

Con esta técnica de recomendaciones basadas en contenido, para poder realizar las recomendaciones tan solo necesitamos una descripción de las características de los ítems y un perfil de usuario que describa los intereses del mismo o sus preferencias para determinadas

características de los ítems. Con todo esto el sistema tan solo tendrá que buscar los ítems que se adapten mejor a las preferencias del usuario. La gran ventaja de este método es que no es necesario una comunidad de usuarios o una historia de puntuaciones, este sistema es capaz de generarlas aunque solo haya un usuario.

A pesar de que los proveedores nos proporcionan descripciones y características de los productos, el principal inconveniente de esta técnica es la extracción de las características cualitativas subjetivas. En un dominio de calidad y gusto, por ejemplo, las razones por la que a una persona le gusta un producto no están relacionadas con las características del mismo y se pueden basar en una impresión subjetiva del diseño exterior del producto.

Estos sistemas vamos a dividirlos en una serie de pasos conceptuales que mostramos a continuación:

- Extracción del conocimiento, en este paso determinamos como extraer las propiedades de los ítems. Estas características las podemos sacar a partir de la descripción que nos dé el proveedor, por ejemplo, para un libro pueden ser el autor, género, precio, etc.

Además de estas propiedades puede existir otra categoría que sean las palabras clave, estas tienen especial interés en recomendaciones basadas en texto, en estos sistemas lo más importante no es la meta-información (autor, editor) sino el contenido del mismo, por ejemplo, en un email, noticias o libros.

Aquí es interesante que el sistema lea el contenido del mensaje y extraiga automáticamente las palabras clave para poder realizar las recomendaciones, para realizar esto los documentos se van a describir usando el formato de codificación TF-IDF, la idea es que los documentos se codifican como vectores multidimensionales en un espacio Euclideo y para calcular la similitud entre dos elementos vamos a usar la media del coseno que forman sus vectores. Esta técnica es conocida como modelo de espacio vectorial y esta descrita en (Salton, Wong, Yang, 1975).

- Generar perfil de usuario, determinamos como generar este perfil, esto se puede realizar de dos maneras diferentes: preguntándole explícitamente al usuario que gustos tiene, autores preferidos, rango de precios, etc. Generalmente estas

características coinciden con las características de los ítems. La otra manera sería decirle que puntuara una serie de ítems para poder extraer que gustos tiene.

- Obtener la recomendación, una vez que tenemos tanto las descripciones de los ítem como el perfil de usuario ya podemos realizar la recomendación, aquí vamos a tener dos formas de hacerlo:
 - El método más simple para calcular recomendaciones consistiría en coger una de las características que nos ha dado el usuario al preguntarle explícitamente y compararla con la del ítem a recomendar.
 - El siguiente método es más sofisticado y es el que suelen usar, está basado en recomendar ítems que son similares a otros que le gustaron al usuario anteriormente. La predicción para un ítem i que todavía no ha sido calificado se realiza buscando los k -vecinos más próximos a i que hayan sido puntuados por el usuario, si por ejemplo cuatro de los cinco ítems más comunes fueron puntuados positivamente entonces el sistema dirá que i tiene una probabilidad alta de que le guste.

Para calcular estos vecinos podemos usar el método KNN (*K-nearest-neighbors*) (Fix y Hodges, 1951) este método puede combinarse con el que vimos anteriormente TF-IDF y la medida del coseno para calcular la similitud entre ítems.

En cuanto a sistemas que usen este paradigma para recomendar tenemos por ejemplo el agente software *Syskill & Webert* (Pazzani, Muramatsu, Billsus, 1996; Pazzani y Billsus, 1997) que es un asistente de navegación web que usa las puntuaciones pasadas que ha dado el usuario a sitios web para predecir si el usuario va estar interesado en los enlaces de una página web.

1.2.3 Recomendadores basados en conocimiento

Como hemos visto hasta ahora en recomendaciones colaborativas y basadas en contenido cada una tenía sus ventajas y sus inconvenientes. Los primeros su principal ventaja es que su uso está muy extendido pero por el contrario solo usan las puntuaciones y ningún tipo de conocimiento adicional para generar las recomendaciones. Basado en contenidos sí que usan diferentes fuentes de conocimiento como el género, palabras clave, etc. para generar sus recomendaciones.

El problema de estos métodos es que existen determinadas situaciones en las cuales estos enfoques no son adecuados, por ejemplo, cuando compramos una casa o un ordenador que son cosas que no se hacen muy frecuentes, para una casa si usamos un sistema colaborativo nos dará malos resultado ya que seguramente existan pocas puntuaciones y en el caso del ordenador si tenemos un sistema basado en contenido y puntuaciones de hace cinco años, tampoco dará buenos resultados ya que los ordenadores puntuados serán muy antiguos.

Para solventar estos problemas surgen los sistemas de recomendación basados en conocimiento, este sistema podría ser visto como un tipo particular de sistema de recomendación basado en contenido. En estos sistemas las recomendaciones se calculan de forma independiente a las calificaciones de los usuarios: ya sea en forma de similitudes entre las necesidades del cliente y los artículos o en función de reglas de recomendación explícitas.

Según (Burke, 2000) una de las principales características de estos sistemas es que no solo valen para el proceso de filtrado sino que además guían al usuario de una forma personalizada hacia los objetos de interés o útiles en un espacio grande de posibles opciones o producen esos objetos como salida.

Los dos tipos básicos de sistemas de recomendación basados en conocimiento que se consideraran serán los basados en restricciones y los basados en casos.

El primer paso en este tipo de sistemas es elegir la representación del conocimiento y el razonamiento, esto dependerá del tipo de sistemas que usemos:

- Basado en restricciones, si usamos este enfoque el problema de una recomendación basada en restricciones puede ser representada como un problema de satisfacción de restricciones por sus siglas en inglés CSP, estos problemas aplicados a los sistemas de recomendación son descritos mediante una tupla con los siguientes elementos:
 - V , donde este valor representa un conjunto de variables, estará formado a su vez por dos conjuntos: V_C que representaran las restricciones de compatibilidad y que describirá todo los posibles requerimientos del usuario (precio máximo [0...100], tipos de película [Acción, ciencia ficción...] etc.) y otro conjunto V_{PROD} , que nos describe las características de los productos (precio, tipo de película, etc.).

- C , este valor representa un conjunto de restricciones que definen que ítems deben de ser recomendados a un usuario en cada situación, está formado a su vez por tres tipos diferentes:

C_R son las restricciones de compatibilidad, define cuales son las instancias permitidas de las propiedades del producto, por ejemplo, si se requieren fotos de gran tamaño, el precio máximo aceptado debe ser superior a 200.

C_F son las condiciones de filtro, define las relaciones entre las propiedades del cliente y las del producto, por ejemplo, las fotografías de gran tamaño requieren resoluciones superiores a 5 mega pixel.

Finalmente C_{PROD} que son las restricciones del producto, define toda la gama de productos disponibles y para cada producto el valor de todas sus características.

- Un conjunto REQ que codificara los requerimientos del usuario, es básicamente la consulta que hace el usuario al sistema.
- Por último un conjunto RES que es una solución válida al CSP en caso de existir.

Una solución a un CSP corresponde en asignar un valor a cada variable V de forma que todas las restricciones sean satisfechas. En nuestro caso cada solución del CSP consistirá en una recomendación consistente.

- Basados en casos, los ítems se recuperan usando medidas de similitud que describan con que grado una propiedad del ítem coincide con alguno de los requerimientos que el usuario nos ha ofrecido. Para calcular la similitud de un ítem a los requisitos del usuario se utiliza la llama distancia de similitud (McSherry, 2003):

$$similitud(p, REQ) = \frac{\sum_{r \in REQ} w_r * sim(p, r)}{\sum_{r \in REQ} w_r}$$

Donde: p es el ítem que queremos averiguar, REQ es el conjunto de requerimientos del usuario, w_r es un peso que le damos al requerimiento r del usuario y $sim(p, r)$ nos dice la similitud del requerimiento r respecto al ítem p , esta es una fórmula que se calcula de la siguiente manera:

$$sim(p,r) = 1 - \frac{|\emptyset_r(p) - r|}{\max(r) - \min(r)}$$

Donde, $\emptyset_r(p)$ es el valor del atributo r del ítem.

Con lo cual el proceso de recomendación se basa en que para cada producto p calcular su similitud y mostrar al usuario los que tengan este valor más alto.

Una vez hemos visto como representar el conocimiento, vamos a ver el flujo de iteración en estos sistemas, según (Jannach, 2011) se resumen en los siguientes pasos:

1. El usuario especifica sus preferencias, por ejemplo, a través de un formulario web. Estos formularios pueden ser generales o personalizados para el usuario. Otros sistemas para obtener las preferencias del usuario prefieren utilizar preguntas /respuestas, estas preguntas se pueden realizar todas a la vez o de una manera incremental.
2. Cuando se ha recogido suficiente información sobre las necesidades o preferencias del usuario, entonces se le presentan una serie de ítem que coinciden con sus preferencias. Opcionalmente, el usuario puede pedir una explicación de porqué se le ha recomendado un determinado ítem.
3. El usuario puede revisar sus preferencias o preferencias, por ejemplo, para ver alternativas o reducir el número de elementos que han coincidido.

El primer ejemplo de este tipo de sistemas lo encontramos en el sistema *VITA* (Felfernig *et.al.*, 2007) se trata de un sistemas basado en restricciones que recomienda productos financieros a los usuarios y que simula un agente de ventas. Sigue el esquema que vimos en este tipo de sistemas: captura los requisitos del usuario, los productos financieros tienen una serie de propiedades y las restricciones en las que se definen que productos deberían ser recomendados en según qué tipo de situaciones.

Otro ejemplo lo tenemos en el sistema *Entree* (Burke, 2000) que recomienda restaurantes en la zona de Chicago que se ajusten las necesidades y deseos de los usuarios, además en (Burke, 2000) tenemos una revisión completa de este tipo de sistemas implementados en diferentes ámbitos de aplicación

En la sección 1.1.2 CBR vimos un ejemplo de sistema de recomendación Top Cases (McSherry, 2005), como vimos era un sistema conversacional para recomendar viajes.

1.2.4 Recomendadores híbridos

Hasta ahora, cada enfoque de recomendación explota diferentes fuentes de información y siguen diferentes paradigmas para realizar las recomendaciones. Colaborativos explotan un tipo específico de información (las puntuaciones de los ítems) de un modelo de usuario junto con las recomendaciones de la comunidad para realizar sus recomendaciones, basados en contenidos se basan en las características del producto y en la descripción textual. Basados en conocimiento razonan en modelos explícitos del conocimiento del dominio. Cada uno de estos enfoques tienen sus pros y sus contras, la idea para crear un sistema híbrido es combinar las ventajas de diferentes algoritmos y modelos de recomendación para resolver los puntos débiles que tienen muchos de estos sistemas por separado.

Para describir su funcionamiento y los diferentes tipos que hay utilizaremos la taxonomía realizada por (Burke, 2002) en la cual distingue entre siete diferentes estrategias de hibridación, a su vez para distinguir estas siete categorías utilizaremos la taxonomía realizada por (Jannach, 2011), que las divide a su vez en tres más:

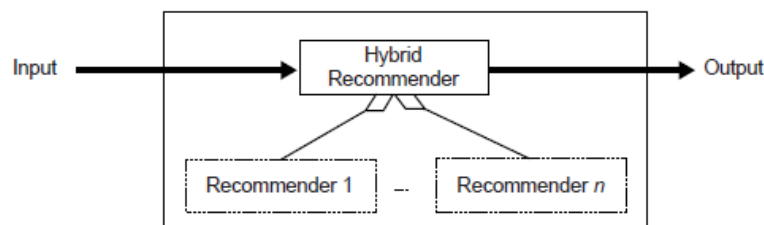
- Diseño monolítico de hibridación, la Figura 1.12 Muestra su diseño. Para conseguir la hibridación incorpora aspectos de diferentes estrategias de recomendación en la implementación de un algoritmo.

Dentro de esta categoría entrarían dos de las estrategias de la categoría de (Burke, 2002)

- *Combinación de características.* Es una técnica que consigue unir recomendaciones basadas en contenido con colaborativas, consiste en tratar la información colaborativa como una característica más asociada a cada ítem y usar técnicas basadas en contenido sobre este conjunto aumentado de datos. En (Basu, Hirsh y Cohen, 1998) tenemos un ejemplo de esta estrategia, se trata de un sistema para recomendar películas en el que combina las característica colaborativas, tales como le gusta o no le gusta, con el contenido de las características de las películas.
- *Aumento de características.* Esta técnica se puede utilizar para integrar varios algoritmos de recomendación. En contraste con combinación de características no se limitan a combinar y pre procesar varios datos de entrada, sino que aplican pasos de transformación más complejos. De

hecho, la salida de un sistema de recomendación que contribuye a aumentar el espacio de características del recomendador actual mediante el preprocesado de sus fuentes de conocimiento. Un ejemplo de esta estrategia la tenemos en el sistema Libra (Mooney y Roy, 1999) es un sistema de recomendación de libros basado en contenido que utiliza los datos que extrae de la página de Amazon para realizar las recomendaciones. Entre las informaciones que extrae están incluidos los autores, relaciones y títulos que es información que genera Amazon de su sistema de recomendación colaborativo. Con lo cual a la hora de realizar la recomendación el sistema no solo va a usar las propiedades del producto recomendado como título, género, etc. sino que además va a usar esa información de Amazon.

Figura 1.12 Diseño monolítico de hibridación (Jannach, 2011)

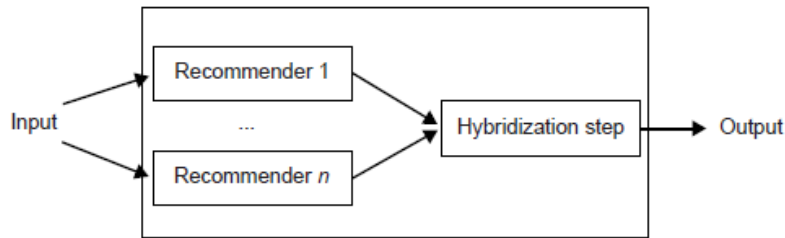


- Diseño de hibridación paralelo, ver Figura 1.13. Este enfoque utiliza paralelamente diferentes recomendadores y utiliza un mecanismo de hibridación específico para agregar las salidas de cada uno de ellos. Siguiendo con la taxonomía de (Burke, 2002) tendríamos:
 - *Mixtos*. Esta estrategia combina los resultados de diferentes sistemas de recomendación al nivel de la interfaz de usuario, en la cual los resultados de diferentes técnicas se le presentan juntas al usuario. Un ejemplo de esta estrategia lo tenemos en el sistema PTV (Smyth y Cotter, 2000) es un sistema de recomendación de programas de televisión, usa tanto técnicas basadas en contenido como colaborativas y el sistema le presenta al

usuario juntas tanto las recomendaciones basadas en contenido como las colaborativas.

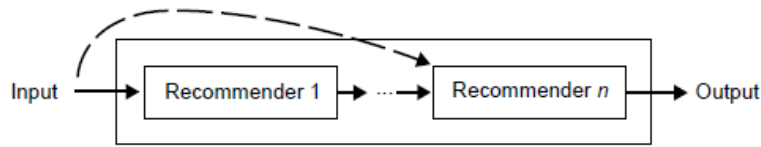
- *Ponderados*. Este enfoque se basa en que la puntuación de un determinado ítem recomendado es calculado a partir de los resultados de todas las técnicas de recomendación disponibles en el sistema, por ejemplo, el más simple sería una combinación lineal de las puntuaciones de recomendación. Un ejemplo de esta estrategia lo podemos ver en Moodvie (Garrido, Peña, Salmerón, 2011) este sistema de recomendación de películas basándose en el temperamento y estado de ánimo del usuario y el género de las películas, en una de sus etapas de filtrado utiliza esta técnica en la cual para obtener la puntuación de una película combina las tres puntuaciones que le han dado tres estrategias basadas en conocimiento.
- *Conmutación*. El sistema usa algún criterio para decidir entre qué sistema de recomendación usar en una situación específica, por ejemplo, para resolver el problema del arranque en frío, un sistema de recomendación híbrido que use técnicas basadas en conocimiento y colaborativas podría inicialmente usar recomendaciones basadas en conocimiento hasta que hubiera suficientes datos de puntuación disponibles. Un ejemplo de esta estrategia la tenemos en el sistema *NewsDude* (Billsus y Pazzani, 1999) se trata de un sistema de recomendación de artículos de noticias que para realizar las recomendaciones tiene dos estrategias basadas en contenido y otra colaborativa, el sistema inicialmente trata de usar la primera estrategia basada en contenido si esta no obtiene ningún resultado cercano entonces pasa a usar la estrategia colaborativa, si esta no ofrece resultados fiables entonces se utiliza la otra estrategia basada en contenido.

Figura 1.13 Diseño de hibridación paralelo (Jannach, 2011)



- Diseño *pipeline* de hibridación, ver Figura 1.14. Esta técnica implementa un proceso por etapas en el que varias técnicas construidas de forma secuencial entre si antes de que la final produzca recomendaciones para el usuario. Aquí hay dos enfoques a su vez, por un lado un componente anterior puede preprocesar los datos de entrada para construir un modelo que sea explotado por la siguiente etapa o bien entregar una lista de recomendaciones para un refinamiento adicional. Siguiendo con la taxonomía de (Burke, 2002) tendríamos:
 - Cascada. En esta técnica, una técnica de recomendación se emplea primero para producir un primer ranking grueso de los candidatos y después una segunda técnica refina la recomendación a partir del conjunto de candidatos proporcionados por el primero.
 - Meta-nivel. Se basa en utilizar el modelo generado por un recomendador como entrada de otro. Este difiere del de aumento de características: en uno basado en características, se utiliza un modelo aprendido para generar una serie de características que serán usadas como entrada de un segundo algoritmo, en este sin embargo todo el modelo se convierte en la entrada del siguiente. Un ejemplo de esta estrategia lo tenemos en el sistema *FAB* (Balabanovic y Shoham, 1997) es un recomendador de noticias online, utiliza un recomendador basado en contenido para construir los modelo de usuario basándose en la categoría y en el grado de interés que tiene el usuario hacia esta categoría. Después para realizar la recomendación usa un modulo colaborativo que encuentra a los usuarios más parecidos basándose en el modelo de usuario.

Figura 1.14 Diseño *pipeline* de hibridación (Jannach, 2011)



1.3 Explicaciones en sistemas de recomendación

En esta sección analizaremos primero cuales son los objetivos de proporcionar explicaciones en un sistema de recomendación, este es un aspecto importante ya que todos los artículos que hemos revisado su objetivo es precisamente la incorporación de explicaciones y ver cómo afectan a alguno de estos objetivos. Después veremos los diferentes tipos de explicaciones que hay implementados en los sistemas actuales de recomendación.

Para explicar los diferentes paradigmas de clasificación de las explicaciones que existen en este tipo de sistemas, vamos a usar la taxonomía descrita por (Friedrich y Zanker, 2011), en este artículo nos presenta varias formas de clasificar las explicaciones en los sistemas de recomendación:

- *Categoría del modelo de razonamiento.* El modelo de razonamiento describe como las explicaciones son generadas de forma que hagan al sistemas más transparente, esta categoría a su vez se divide en dos mas:
 - *Explicaciones de caja blanca.* Revelan al usuario al menos parcialmente el modelo de razonamiento y su contenido, de una forma parecida a que lo hacían las explicaciones de tipo *how?* en los sistemas expertos.
 - *Explicaciones de caja negra.* Generan justificaciones que argumentan porque la recomendación específica es plausible o debería ser de interés aunque el modelo de razonamiento no tuvo en cuenta esas proposiciones cuando calculaba la recomendación.
- *Basadas en el paradigma de recomendación.* Estas al igual que los tipos de sistemas de recomendación que vimos en la sección anterior utilizan el mismo paradigma para generar la explicación, pero en este caso en vez de recomendar ítems explican porque uno en especial es de utilidad para el usuario. Un aspecto importante a tener en cuenta y que no ha que confundir es que nosotros podemos haber usado un paradigma colaborativo para generar la recomendación, pero para

generar la explicación podemos haber usado uno basado en contenido, con lo cual el algoritmo de recomendación no tiene que ser el mismo que genera la explicación. Los diferentes tipos de explicaciones basadas en el paradigma de recomendación son los siguientes:

- *Colaborativas*
- *Basadas en contenido*
- *Basadas en conocimiento*
- *Categoría de información.* Esta se refiere a la categoría de información que se explota para generar la explicación. Aquí se diferencia tres aspectos diferentes de entrada:
 - *Modelo de usuario.* En estas el sistema podría presentar argumentos basados en puntuaciones conocidas del usuario, preferencias o características demográficas, por ejemplo, “Tu puntuación media es 3.5”.
 - *Ítem recomendado.* La explicación se centra en el ítem recomendado, concentrándose en determinadas características que tiene el mismo, por ejemplo, “La película propuesta gana tres Globos de Oro y dos nominaciones a los Oscar”.
 - *Alternativas.* Las explicaciones argumentan a favor o en contra de alternativas al ítem recomendado.

Además los tres tipos anteriores se pueden combinar juntos para dar lugar a explicaciones mucho más complejas.

De los tipos de clasificación de las explicaciones que acabamos de describir nos hemos decantado por la segunda, ya que nos parece la más intuitiva y fácil de entender para alguien que tenga conocimiento en sistemas de recomendación. Para cada uno de los diferentes paradigmas de recomendación que hay, en la siguientes subsecciones presentamos una revisión actual de los diferentes sistemas de recomendación que hayan implementado este tipo de explicaciones en sus sistemas.

1.3.1 Objetivos de la recomendación

Un aspecto importante a la hora de incorporar explicaciones a un sistema de recomendación son los objetivos, cual es el propósito de generar explicaciones en este tipo de

sistemas, en la sección de explicaciones de sistemas expertos ya vimos algunos, el problema es que ninguno de los sistemas los evaluaba realmente tan solo los citaban. La investigación en este campo ha ido avanzando y ha aportado nuevos objetivos y criterios, a continuación mostramos una lista de los objetivos basándose en los trabajos de (Jannach, 2011; Tintarev, 2007), en especial este último, en el cual crea una lista de objetivos basándose en los sistemas de recomendación que incorporaron explicaciones y cuáles fueron los criterios que se usaron para poder evaluar su impacto, estos objetivos son los siguientes:

- *Transparencia.* Las explicaciones apoyan la transparencia de las recomendaciones, con el objetivo de proporcionar información que el usuario pueda comprender sobre el razonamiento utilizado para generar una recomendación específica. Ofrece información de porqué un tema es elegido sobre otro. Todo esto logra que el usuario pueda entender cómo funciona el sistema. (Cramer *et al.*, 2008) ha investigado los efectos de la transparencia en otros criterios de evaluación como la persuasión, confianza, etc.
- *Validez.* Las explicaciones se pueden generar para permitir que un usuario verifique la validez de una recomendación, por ejemplo, “Yo te recomiendo este tipo de coche, ya que tienes cuatro hijos y el cocheX de la marca Y tiene siete plazas”. Esto permite al usuario corregir el razonamiento. Este criterio además está muy unido a la transparencia haciendo que en muchos casos sean identificados juntos, aunque existen casos en los que explicaciones transparentes no pueden ser validadas, por ejemplo, un sistema recomendador que utilice un rating, presente una transparencia en como calculo esa recomendación pero luego no permite al usuario modificar esos rating con lo que no esté de acuerdo.
- *Confianza.* Para (Jannach, 2011) es visto como un mecanismo que reduce la complejidad de la toma de decisiones humanas en situaciones de incertidumbre, por ejemplo, “El fármaco Kural, ha curado a miles de personas con su enfermedad, por lo tanto, este medicamento le ayudara”. Para (Tintarev, 2010) está relacionado con que el usuario crea que el proceso de recomendación es correcto e imparcial, estudios (Felfernig y Gula, 2006; Sinha y Swearingen, 2002) indican que la confianza esta unido en muchos aspectos a la transparencia, de modo que si hacemos sistemas recomendadores transparentes y una cierta interacción del

usuario con el sistema lograremos aumentar la confianza en el mismo, esto unido al estudio de (Chen y Pu, 2002) que dice que sistemas que den más confianza al usuario será más probable que vuelvan a ser usados por el mismo. Otro aspecto importante es la interfaz del sistema de recomendación ya que esta puede afectar seriamente a su credibilidad.

- *Persuasión.* Es considerado como persuasivo si el sistema está diseñado para cambiar la actitud o conducta de una persona de una manera predeterminada, por ejemplo, una recomendación que hable solo de los aspectos positivos de un producto y oculte los negativos, de forma que tratan de obtener beneficios para el sistema en lugar de para los usuarios. Es importante tener en cuenta que un uso excesivo de la persuasión puede ser contraproducente una vez los usuarios se den cuenta de que han tratado o han comprado productos que realmente no querían.
- *Efectividad.* Explicaciones para mejorar la efectividad ayudan al cliente a descubrir sus preferencias y tomar decisiones que maximicen su satisfacción respecto al producto seleccionado, básicamente, ayudarle a que tomen mejores decisiones. Las Explicaciones no solo valen para convencer a un usuario de que compre un determinado producto sino que también pueden usarse para ayudarle a tomar decisiones. Pero la efectividad es altamente dependiente de la precisión del algoritmo de recomendación ya que los usuarios no tomaran decisiones correctas si las recomendaciones son defectuosas.
- *Eficiencia.* Se refiere a la capacidad de apoyo del sistema a un usuario para reducir esfuerzo de la toma de decisiones, por lo general tratan de reducir el tiempo en la toma de decisiones.
- *Satisfacción.* Las explicaciones tratan de mejorar la satisfacción general resultante del uso del sistema de recomendación, por ejemplo, aumentar la tasa de retorno de los usuarios.
- *Relevancia.* Las explicaciones se pueden proporcionar para justificar porque un usuario necesita cierta información adicional, este objetivo es de especial interés en sistemas de tipo conversacional.
- *Comprensibilidad.* Los recomendadores no pueden estar nunca seguros sobre el conocimiento de sus usuarios. Explicaciones dirigidas a apoyar la comprensión

del usuario relacionan los conceptos conocidos del usuario con los conceptos empleados por el recomendador.

- *Educación.* Las explicaciones pueden tener como objetivo instruir a los usuarios para ayudarles mejor a entender el dominio de un producto, de forma que realicen mejores decisiones de compra.

Hay que tener en cuenta que muchos de estos objetivos están relacionados entre sí, por ejemplo, un aumento de la transparencia seguramente hará que la confianza del usuario en el sistema aumente, por el contrario, un aumento de la persuasión es probable que produzca una disminución en la confianza del sistema.

1.3.2 Explicaciones colaborativas

Según (Jannach, 2011) uno de los principales problemas que tienen los filtros colaborativos es que son incapaces de justificar porque un determinado producto es adecuado para un cliente o porque un producto no cumple con las necesidades del cliente. Estos sistemas siguen la idea de seguir el proceso de recomendación boca en boca que realizamos las personas. Una de las principales tareas de las explicaciones será justificar como este sistema de boca en boca trabaja.

Generalmente estos métodos utilizan algoritmos muy sofisticados, en teoría la ejecución de esos algoritmos debería servir como una explicación de la salida. Pero en realidad esa explicación tiende a contribuir a la confusión de los clientes y tiene un impacto negativo. El arte del diseño de estos sistemas será proporcionar un nivel de abstracción adecuado al nivel de los clientes.

El único trabajo de investigación que existe en el ámbito de explicaciones colaborativas y a su vez uno de los más influyentes en el ámbito de las explicaciones en sistemas de recomendación es propuesto por (Herlocker, Konstand y Riedl, 2000), este artículo nos propone un modelo para poder generar explicaciones siguiendo la categoría del modelo de razonamiento de la taxonomía propuesta por (Friedrich y Zanker, 2011), que hemos visto al principio de la sección 1.3:

- Explicaciones de caja blanca, aquí nos propone un modelo para poder generar explicaciones que describan el modelo conceptual que sigue un algoritmo colaborativo, según los autores cualquier algoritmo de este tipo puede ser dividido

en tres pasos conceptuales y para cada uno de estos generar explicaciones acordes al paso, este tipo de explicaciones se van a centrar en la justificación del proceso de razonamiento seguido por el algoritmo. Los tres pasos conceptuales junto a sus explicaciones son las siguientes:

- Puntuaciones de los productos que ha hecho el cliente. Un aspecto fundamental es mostrar al cliente algún tipo de información sobre la fuente de las puntuaciones. Algunos sistemas son claros y solo utilizan como fuente las puntuaciones que hayan utilizado para los productos, pero existen otros sistemas que consideran otras informaciones que están basadas en observaciones sobre el cliente y desde las cuales deducen compartimientos. Por ejemplo, un sistema observa que un usuario está interesado en libros de ciencia ficción, con lo cual deduce que también podría estar interesado en películas de ciencia ficción. Además los clientes pueden estar interesados en como de influyentes son algunas de sus puntuaciones.
- Localizar a los clientes con gustos similares. Desde el punto de vista del cliente será muy importante saber si los vecinos que ha utilizado el sistema tienen gustos similares. Proporcionar esta información ayudara al cliente a evaluar la calidad de la predicción y para especificar que vecinos son más aceptables para la predicción de las puntuaciones.
- Calculo de puntuaciones mediante la combinación de las puntuaciones de los vecinos. Este último paso consiste en explicar los datos y el proceso de como tomar las puntuaciones de los vecinos y agregarlos en una puntuación final. En este paso lo más recomendable para generar la explicación es combinar este paso junto a la idea del paso anterior y mostrarle al usuario, por ejemplo, en forma de una interfaz que muestre como ha puntuado cada uno de sus vecinos el ítem recomendado.
- Explicaciones de caja negra, en este caso puede que no estemos interesados en mostrarle al usuario el modelo conceptual del sistema, y quizás nos convenga mas proporcionar explicaciones cortas, como justificaciones o razones para confiar en la recomendación pero sin mostrar el modelo que seguimos. Una técnica muy

usada es usar el rendimiento pasado del recomendador, por ejemplo, “Este sistema ha hecho predicciones correctas el 80% del tiempo cuando recomienda ítems similares a este”.

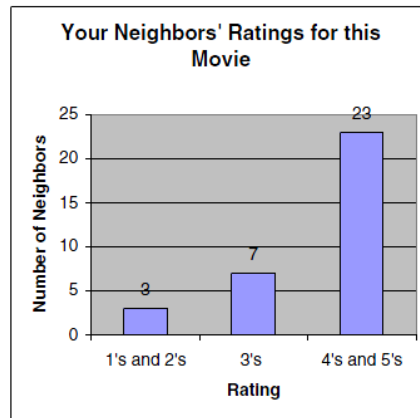
Otro aspecto importante del artículo son los resultados que extrae de examinar una veintena de interfaces gráficas de explicación en el contexto de *MovieLens*. A los clientes se les pregunta qué valoraran de uno al siete la probabilidad de que fueran a ver una película recomendada una vez se le haya presentado la explicación, también destacar que se evaluó respecto al caso en el que se no se proporcionara ningún tipo de explicación para poder hacer comparaciones. Los resultados fueron los siguientes:

- Las explicaciones que dieron mejores resultados fueron las basadas en las puntuaciones de los vecinos, Figuras 1.15 y 1.16. El histograma fue la explicación que mejor valoraron los usuarios.

Figura 1.15 Explicación que muestra los rating y vecinos en forma de la tabla (Herlocker, Konstant y Riedl, 2000)



**Figura 1.16 Explicación que muestra los rating y vecinos en forma de histograma
(Herlocker, Konstam y Riedl, 2000)**



- Las explicaciones que usan el rendimiento pasado del sistema como justificación fueron las que obtuvieron la segunda mejor puntuación, la explicación exactamente es la siguiente, “*MovieLens* ha predicho correctamente para ti el 80% del tiempo en el pasado”.
- Las explicaciones que usan el contenido de la película para tratar de convencer, por ejemplo, tus actores favoritos o mostrar películas similares que hayas valorado para generar la explicación también obtuvieron muy buenos resultados.
- Las interfaces de explicación que estén mal diseñadas disminuyen la disposición de los usuarios a ir a ver la película recomendada, incluso comparándola con el caso en que no mostremos ninguna explicación.
- Demasiada información tiene efectos negativos, por ejemplo, a la Figura 1.16 se le añadió información sobre la proximidad de los vecinos y se presentado en forma de grafo, esto hizo que el resultado empeorara y fuera una de las peor valoradas.

1.3.3 Explicaciones basadas en contenido

En estos sistemas el conocimiento sobre los ítems se da, por ejemplo en forma de un catalogo de ítems con las descripciones de los mismos. Las recomendaciones se calculan determinando aquellos ítems que son más similares a los ítems que sabemos que le gustan al usuario, con lo cual podemos decir que los algoritmos basados en contenido consideran similitud

entre ítems, basándose en la puntuación de los usuarios pero considerando las propiedades de los ítems. Las explicaciones basadas en contenido siguen este mismo esquema, están basadas en las propiedades de los ítems.

En este contexto sí que existen diversos trabajos de investigación que tratan las explicaciones, es más la gran parte de los trabajos sobre explicaciones en sistemas de recomendación usan este paradigma para sus modelos, a continuación presentaremos unos cuantos trabajos que nos han parecido los más representativos dentro de este ámbito.

El primero de estos es propuesto por (Vij, Sen y Riedl, 2009), es desarrollado por uno de los autores del artículo que vimos en la sección de explicaciones colaborativas y al igual que aquel este también está aplicado en el contexto de *MovieLens*.

Los autores nos describen un sistema para generar explicaciones usando las etiquetas que tienen las películas, estas etiquetas las proporcionan los usuarios, esto tiene una ventaja ya que no se necesitan expertos que las creen y mantengan. Sin embargo también tienen sus inconvenientes ya que muchas veces son redundantes o tienen poca calidad.

Estas etiquetas van a servir para añadir características a las películas de forma que las recomendaciones estarán basadas en buscar estas mismas etiquetas en otras películas y ver si le gustó al usuario, de forma que la generación de la explicación estará basado en mostrar al usuario información sobre sus antiguas valoraciones de las etiquetas. Para diseñar estas etiquetas va haber dos aspectos clave:

- Relevancia de la etiqueta. Es el grado con el que una etiqueta describe un ítem, para calcular este valor determinamos una correlación entre la preferencia del usuario por esa etiqueta y su preferencia por la película, esto quiere decir, a los usuarios que les gusta una determinada etiqueta también les gusta el ítem asociado.
- Preferencia de la etiqueta. Es el sentimiento del usuario hacia una etiqueta, para estimar este valor se calcula la puntuación media de los usuarios que han dado una puntuación a una película con esa etiqueta.

En cuanto a los resultados del artículo, los autores proponen cuatro interfaces de explicación y a los usuarios se les realizó una encuesta en la que querían evaluar los siguientes aspectos:

- Justificación. La explicación me ayudo a entender la predicción que ha hecho el sistema.
- Efectividad. La explicación me ha ayudado a determinar cuánto disfrutare la película.
- Compatibilidad con estado de ánimo. La explicación me ayudado a decidirme si esta película estará bien para mi estado de ánimo actual.

Para cada una de las preguntas anteriores el usuario puede dar una puntuación del uno al cinco. La interfaz que mejor resultado dio en todos los aspectos es la que se muestra en la Figura 1.17, esta interfaz nos muestra la relevancia y la preferencia que ha dado el usuario a las etiquetas, cada una de estas etiquetas esta ordenada por la relevancia de las mismas.

Figura 1.17 Explicación que nos muestra la relevancia y preferencia que tiene un usuario hacia una etiqueta (Vij, Sen y Riedl, 2009)



El siguiente trabajo que vamos a ver es el propuesto por (Cramer *et al.*, 2008), se trata de un estudio que nos muestra cómo influye la transparencia en la confianza del usuario y su aceptación a través de explicaciones basadas en contenido. Este sistema recomienda obras de arte del museo *Rijksmuseum* en Ámsterdam, las recomendaciones se generan de la siguiente manera:

- Todas las obras están anotadas semánticamente por los expertos del museo, esto va permitir extraer una serie de características de la misma.
- Se crea un perfil de usuario donde a partir de las puntuaciones que él da a las obras se extraen las características de las mismas que le gustan o no.

- Una vez tenemos este perfil podemos adaptar las recomendaciones a cada usuario, de forma que si sale una nueva obra, en función de sus características podemos predecir si le gustara.

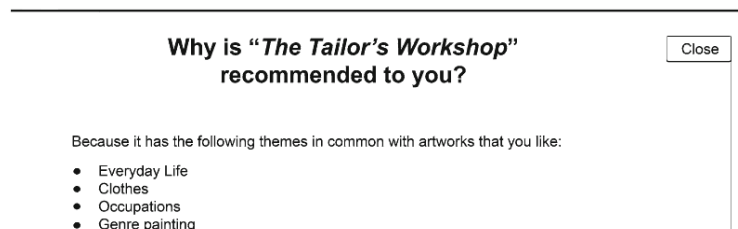
El estudio surge de la premisa que la transparencia influye en la confianza, comprensión y aceptación del sistema de recomendación. Para averiguar esto, proponen tres experimentos:

- No ofrecer ningún tipo de transparencia, esto se logra no ofreciendo ninguna explicación.
- Una condición de transparencia, se muestra la obra y debajo de la misma un enlace que conduce a una explicación de ¿por qué? el usuario disfrutara la obra, una vez el usuario pulse en enlace *Why?* se abrirá una ventana donde se le dará una explicación, esta está basada en una lista con una serie de características de la obra recomendada tiene en común con otras obras que el usuario voto positivamente en el pasado, en las Figuras 1.18 y 1.19 se muestra las interfaces generadas.

Figura 1.18 Resultado de una recomendación en el sistema diseñado por (Cramer *et al.*, 2008)



Figura 1.19 Explicacion que justifica ¿Por qué? una obra le gustara al usuario (Cramer *et al.*, 2008)



- Una condición de índice de confianza, debajo de la imagen, la explicación muestra un porcentaje que indica la confianza que tiene el sistema en cuanto le va gustar la recomendación al usuario, en la Figura 1.20 se muestra esta interfaz.

Figura 1.20 Explicación con índice de confianza (Cramer *et al.*, 2008)



Para la evaluación proponen una serie de entrevistas y cuestionarios en los que tratan de evaluar los siguientes aspectos:

- Transparencia percibida por el sistema. Entiendo en que se basa el sistema para hacer sus recomendaciones.
- Entendimiento del sistema. Cómo funciona el sistema.
- Competencia percibida. Las recomendaciones coinciden con tus intereses.
- Intención de usar el sistema. La próxima vez que esté interesado en una recomendación para una obra, usare el sistema.
- Aceptación del sistema.
- Aceptación de la recomendación.
- Utilidad y facilidad percibidas. Encontré las explicaciones de tipo “¿Por qué?” útiles.
- Confianza. Creo las recomendaciones que hace.
- Necesidad percibida de las explicaciones. Pienso que el sistema debería darme una explicación de ¿Por qué? me recomienda este ítem.

Los principales resultado que ofreció el estudio fueron los siguientes:

- Efectos de la transparencia en cuanto al entendimiento y transparencia percibida. Los participantes en el estudio encontraron la explicación con transparencia hacia

el sistema más transparente que la que no ofrecía. La explicación basada en índice de certeza generaba resultado parecidos a la interfaz que no ofrecía explicación.

- Efectos de la transparencia en la competencia percibida. El hecho de que el sistema ofrece una explicación de transparencia, no hace que aumente la competencia del mismo. Eso se debe, por ejemplo, a usuario que sean muy expertos pueden estar en desacuerdo con las etiquetas que tienen las obras de arte, además pueden pensar que las explicaciones son demasiados simples.
- Efectos de la transparencia en la aceptación del sistema y recomendación. En cuanto a la aceptación en la recomendación, podemos afirmar que ofrecer una explicación de transparencia sí que hace aumentar este aspecto, pero para la aceptación del sistema no.
- Efectos de la transparencia en la confianza del sistema de recomendación. Contrario a la hipótesis inicial, el hecho de mostrar una explicación que justifique no hace que la confianza del sistema aumente respecto a la interfaz que no muestra explicaciones.

El siguiente trabajo que analizaremos es la tesis doctoral de (Tintarev, 2010), de todos los trabajos que hemos analizado este es el más completo y el que ofrece más resultados, se trata de un estudio del impacto de la explicaciones en los sistema de recomendación. Este trabajo se centra sobre todo en estudiar los efectos de personalizar las características de los elementos y en cómo pueden afectar a la eficacia de la explicación, la persuasión y satisfacción de los usuarios.

Las explicaciones que va generar el sistema van a estar basadas en diferentes grados de personalización:

- Básicas. La explicación no está personalizada, no describe características de los ítems.
- No personalizadas, basadas en características. Las explicaciones describen características de los ítems, pero estas propiedades no están adaptadas al usuario.
- Personalizadas, basadas en características. Las explicaciones describen características de los ítems y están adaptadas a los intereses del usuario.

Las hipótesis de las que parte el estudio y quieren ver si son ciertas son:

- Generar explicaciones que estén basadas en personalización será más efectivas que explicaciones que no estén personalizadas y básicas.

- Los usuarios estarán más satisfechos con explicaciones personalizadas en comparación con las que no están personalizadas y básicas.

Para probar esta hipótesis proponen tres experimentos:

- En el primero se propone en un dominio de películas, quieren comprobar si usando características de las películas (actor, director) y personalización en la explicación, esta afecta a la efectividad y satisfacción del usuario. También quieren comprobar cómo la personalización afecta a la persuasión, si le ayudamos a tomar mejores decisiones, entonces acabará comprando/probando menos elementos. Para poder realizar las recomendaciones primero se realiza un formulario para poder extraer las características que más le gustan de las películas:
 - Géneros que le gustan y no le gustan.
 - Características que le gustan de la películas
 - Actores, directores, etc.

Después cada uno de los usuarios se le distribuyó en torno a uno de los diferentes grados de explicaciones que vimos más arriba:

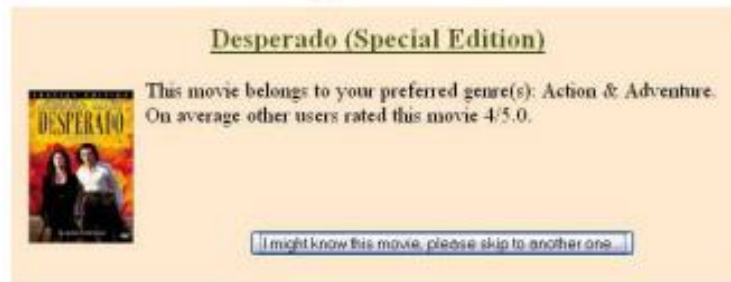
- Básicas. Por ejemplo, “Esta es una de las cien mejores películas de la base de datos del IMDB”, Figura 1.21.

Figura 1.21 Explicación básica en (Tintarev, 2010)



- No personalizadas. En este caso se genera la explicación en torno a una característica que se selecciona aleatoriamente y el género de la misma, por ejemplo, “Esta película pertenece a tu género preferido: Acción y Aventura. La media que han dado los usuarios a esta película ha sido de 4/5.0”, Figura 1.22.

Figura 1.22 Explicación no personalizada (Tintarev, 2010)



- Personalizadas. En este caso se genera la explicación en torno al género y la característica que más le gusta al usuario, por ejemplo, “Aunque esta película no pertenece a ninguno de tus géneros preferidos, es un Documental. Esta película está protagonizada por Ben Kingsley, Ralph Fiennes y Liam Neeson, tus actores preferidos”.

Los resultados del estudio fueron los siguientes:

- El grado de personalización o el usar características de los ítems no incrementa la efectividad, es más, las explicaciones no personalizadas demostraron ser más efectivas.
- Aunque las explicaciones no personalizadas resultaron ser más efectivas, las personalizadas sí que aumentaron la satisfacción de los usuarios.

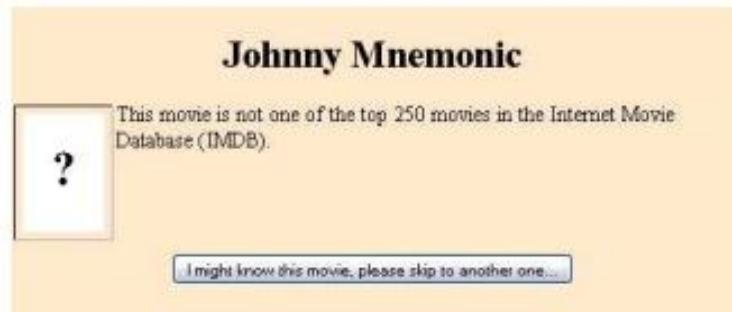
El segundo experimento también se realiza en un entorno de películas, realizan una serie de modificaciones sobre el anterior:

- Eliminan las imágenes para comprobar que influencia tienen.
- La información sobre los géneros es mas detalla y completa.
- La explicación no personalizada describe todos los géneros de la película y ya no los relacionan con los gustos del usuario, además la característica que selecciona se hace de forma aleatoria para el usuario, puede ser cualquiera menos la que haya votado en primer lugar.

Las características que más le gustan al usuario de las películas se realizan igual que en el paso anterior, y también como el experimento anterior a los usuarios se les distribuyo en torno a los diferentes grados de explicaciones:

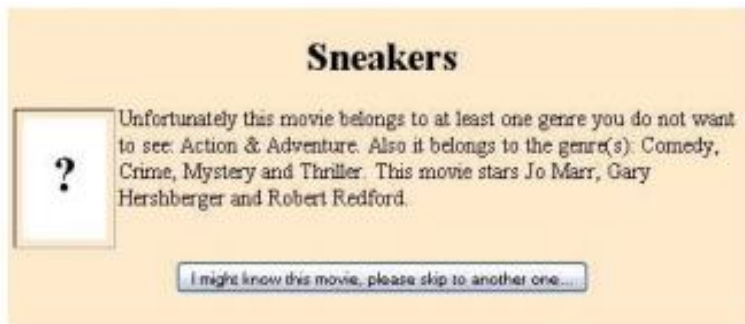
- Básicas. Las mismas que en el paso anterior.
- No personalizadas. Igual que en paso anterior, pero teniendo en cuenta las restricciones, la Figura 1.23 muestra un ejemplo.

Figura 1.23 Explicación no personaliza sin imagen (Tintarev, 2010)



- Personalizadas. Igual que antes, Figura 1.24.

Figura 1.24 Explicación personaliza sin imagen (Tintarev, 2010)



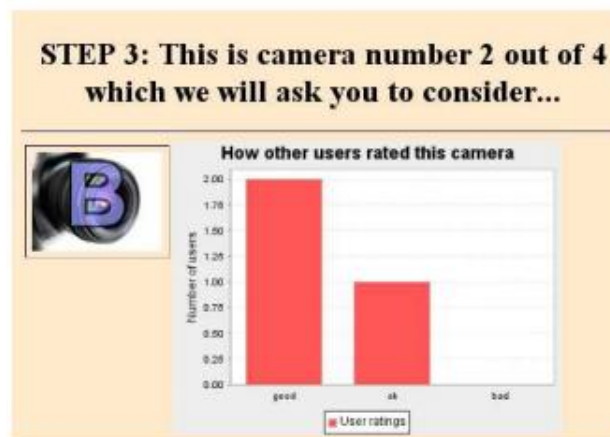
Los resultados del estudio fueron los siguientes:

- Al igual que sucedía con el otro experimento la versión no personalizada muestra una mayor efectividad que la personalizada. También destacar que la ausencia de imágenes no parece que afecte mucho a la efectividad en comparación con otros factores.
- Aunque las explicaciones no personalizadas resultaron ser más efectivas, las personalizadas sí que aumentaron la satisfacción de los usuarios.

Debido a la sorpresa que supuso el hecho de que las explicaciones no personalizadas fueran más efectivas que las personalizadas, los autores quisieron investigar si este fenómeno era particular del dominio de las películas o si por el contrario era reproducible en otros entornos. Para ello seleccionan un entorno de cámaras en el que las características que extraen de las mismas son las siguientes: marca, zoom óptico, precio, resolución, peso, memoria y tipo de cámara. Para extraer las características se usó un formulario como en los experimentos anteriores. El estudio demostró que las propiedades que más tienen en cuenta los usuarios son el tipo de cámara, marca y el precio. Al igual que en los otros experimentos los usuarios se distribuyeron entorno a tres interfaces de explicaciones:

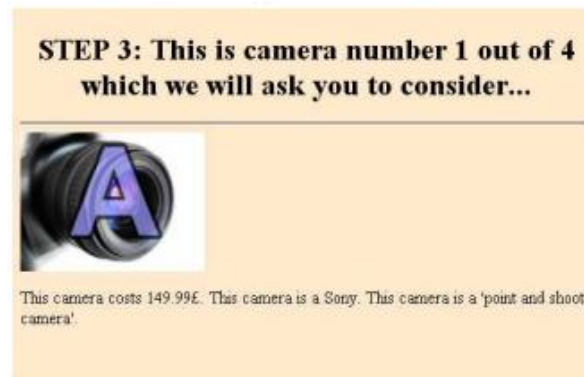
- Básicas. Debido a que en este dominio no hay algo parecido al IMDB para películas, se ha optado por la opción de un histograma al estilo de la Figura 1.16 del trabajo de (Herlocker, Konstant y Riedl, 2000), la diferencia es que este muestra tres opciones, bueno, normal y malo, Figura 1.25.

Figura 1.25 Explicación básica en un contexto de fotografía (Tintarev, 2010)



- No personalizadas. Ejemplo, “Esta cámara ha costado 180 \$. Es una cámara Panasonic”, ver Figura 1.26.

Figura 1.26 Explicación no personaliza (Tintarev, 2010)



- Personalizadas. Por ejemplo, si para un usuario las características de precio, marca y zoom son las más importantes, la explicación debe ser: “Esta cámara cuesta 700\$. Esta cámara es una Nikon. Tienen un zoom óptico de 11”

A parte de las hipótesis iniciales de las que parte el estudio, en este se ha incluido una más: Los usuarios son propensos a sobreestimar sus puntuaciones sobre la cámara en la condición básica respecto a las explicaciones basadas en características (persuasión).

Los resultados fueron los siguientes:

- Se ratifica lo que sucedía en los otros experimentos, las explicaciones personalizadas no son más efectivas que las no personalizadas, además también se desestima la tercera hipótesis ya que los usuarios de las explicaciones básicas no tienden a sobreestimar.
- También se ratifica que los usuarios están más satisfechos con explicaciones personalizadas respecto a las no personalizadas, pero sin embargo no existe diferencia entre las básicas y las personalizadas

El último trabajo que presentaremos es el realizado por (Bilgic y Mooney, 2005), este trabajo es cierto que no solo se centra en explicaciones basadas en contenido sino que también presenta colaborativas, pero debido a que en un principio el sistema de recomendación estaba basado en contenido en su totalidad y el grueso del trabajo se centra en este tipo de explicaciones, hemos preferido meterlo en esta subsección, hecha esta aclaración el foco de este trabajo es generar explicaciones y ver cómo afectan a la efectividad de la recomendación. Este sistema de recomendación se usa para recomendar libros a los usuarios, la versión actual usa un enfoque híbrido para generar las recomendaciones, aunque inicialmente fue concebido como un

recomendador basado en contenido. Para generar las explicaciones se siguen los siguientes pasos:

- Todas las obras están anotadas semánticamente, se extrae: Título, autor, descripción, materia, autores relacionados, obras relacionadas. Cada una de estas características tiene una serie de palabras junto a la frecuencia con la que aparecen.
- Se le propone al usuario que puntué una serie de libros y se le crea un perfil de usuario usando las palabras que tienen las diferentes características de libros que ha votado usando un clasificador Bayesiano ingenuo.
- Después se hacen predicciones basadas en el contenido de los libros, usando las puntuaciones que va dando el usuario y se buscan los vecinos que más se le parezcan y por último se hace la predicción basándose en estos vecinos.

Los estilos de las explicaciones son los siguientes:

- Basadas en palabras clave. Estas explicaciones analizan el contenido de un ítem y encuentran las mayores coincidencia con el perfil de usuario, por ejemplo, para el apartado descripción mirara las palabra que tiene y para cada una de ellas mirara el número de veces que aparece y por último la fuerza, esta mide como de probable es que una palabra aparece en un libro valorado positivamente, ver Figura 1.27. Además presenta una explicación de mayor detalle para saber donde se ha votado esa palabra en otros libros, ver Figura 1.28.

Figura 1.27 Explicación basadas en las palabras clave (Bilgic y Mooney, 2005)

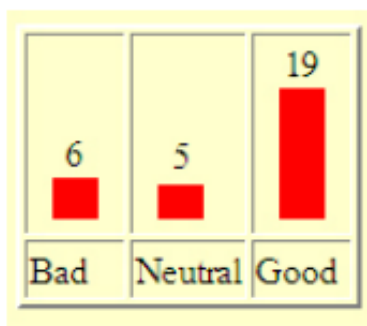
Slot	Word	Count	Strength	Explain
DESCRIPTION	HEART	2	94.14	Explain
DESCRIPTION	BEAUTIFUL	1	17.07	Explain
DESCRIPTION	MOTHER	3	11.55	Explain
DESCRIPTION	READ	14	10.63	Explain
DESCRIPTION	STORY	16	9.12	Explain

Figura 1.28 Explicacion basada en las palabras clave, te muestra las obras donde salen esas palabras (Bilgic y Mooney, 2005)

Title	Author	Rating	Count
Hunchback of Notre Dame	Victor Hugo, Walter J. Cobb,	10	11
Till We Have Faces : A Myth Retold	C. S. Lewis, Fritz Eichenberg,	10	10
The Picture of Dorian Gray	Oscar Wilde, Isobel Murray,	8	5

- Basadas en los vecinos. La explicación es un grafico de barras en el que se representan las puntuaciones de los vecinos, agrupadas por malas, neutrales y buenas, al estilo de Figura 1.16, ver Figura 1.28.

Figura 1.28 Explicación basada en los vecinos (Bilgic y Mooney, 2005)



La metodología seguida para obtener los resultados es la siguiente:

1. Obtener unos ejemplos de puntuaciones del usuario.
2. Calcular recomendaciones. r .
3. Para cada una de las explicaciones. e .
 - a. Presentar r al usuario junto a una explicación e
 - b. Preguntar al usuario que puntué r .
4. Pedir al usuario que pruebe r y lo vuelva a puntuar.

Se parte de las siguientes hipótesis:

- Explicaciones basadas en los vecinos harán que el usuario sobrestime la puntuación de un ítem.

- Las explicaciones basadas en palabras claves permitirán estimar con mayor precisión

Los resultados obtenidos fueron los siguientes:

- La explicación basada en vecinos se comprobó que efectivamente hacia que los usuarios sobrestimarán las puntuaciones.
- Las explicaciones basadas en palabras claves fueron las más efectivas, las puntuaciones dadas en el paso 3 y 4 fueron prácticamente las mismas, esto debe a que este tipo de explicación ayudan a comprender mejor que hay en los ítems previamente puntuados que es similar a los actuales.

1.3.4 Explicaciones basadas en conocimiento

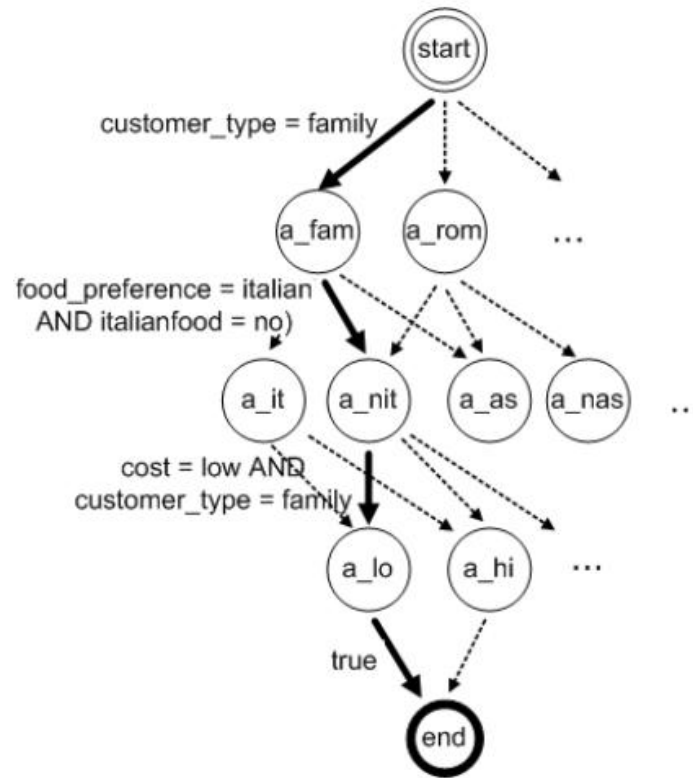
En estos sistemas asumimos que la entrada del motor de recomendación son descripciones de las necesidades o intereses del usuario. El motor de recomendación entonces infiere algún ítem que coincida con las necesidades del usuario. Estos por ejemplo codifican experiencias explícitas de venta tales como que características del ítem ayudan a cumplir un requisito específico del usuario, con lo cual la explicación ira orientada a mostrar al usuario que características del ítem hacen que se cumplan los requisitos del mismo.

El único trabajo que vamos a revisar en esta sección es el propuesto por (Zanker y Ninaus, 2010), en este artículo los autores nos proponen un *framework* para generar explicaciones basadas en conocimiento, este sistema una de sus principales ventajas es que nos permite desacoplar el proceso de razonamiento de las recomendaciones del de la generación de explicaciones, también destacar que es independiente del mecanismo de recomendación.

Las explicaciones que genera son de tipo *Why?*, esto quiere decir que nos explican ¿Por qué? me recomienda un ítem determinado, no se centran en explicaciones de tipo *How?* que son mas técnicas y que quizás no interesen tanto a los usuarios.

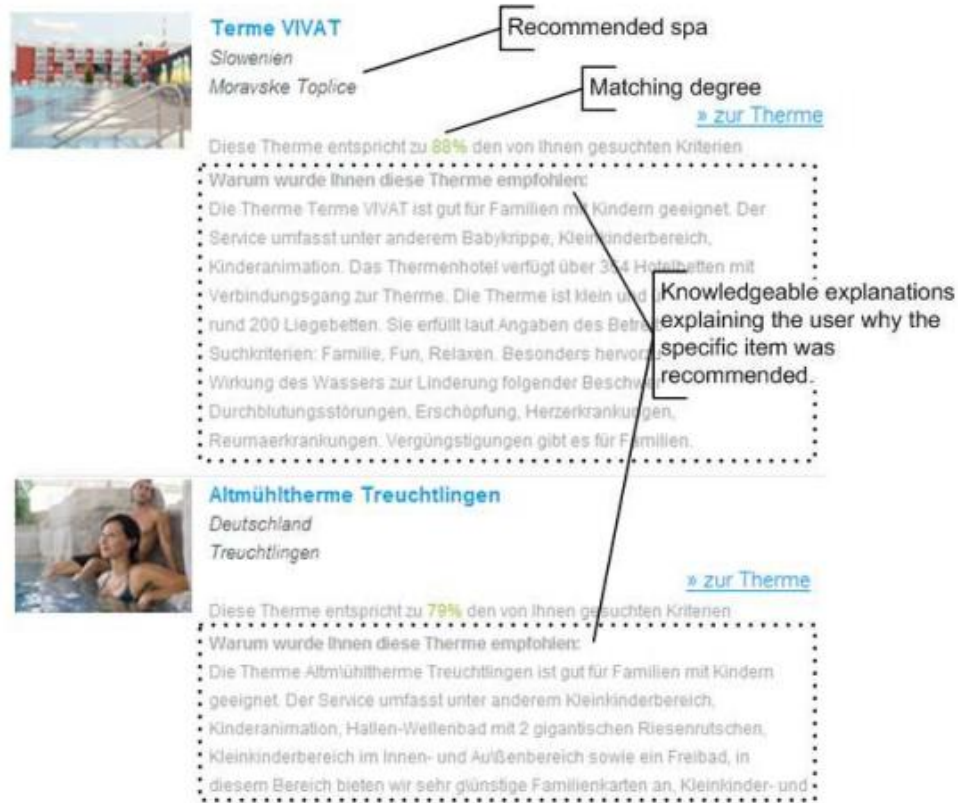
El modelo de explicación viene representando mediante un grafo a cíclico por capas que tienen un nodo inicial y final, ver Figura 1.29.

Figura 1.29 Modelo de explicación propuesto por (Zanker y Ninaus, 2010)



En donde una explicación es una secuencia de argumentos $e = \langle a_1, \dots, a_n \rangle$ donde cada argumento $a \in e$ puede ser una frase textual y e en su conjunto puede ser un texto en lenguaje natural, como se muestra en la Figura 2.30.

Figura 2.30 Explicación completa en (Zanker y Ninaus, 2010)



Con lo cual una explicación va consistir en encontrar un camino que vaya desde el nodo inicial al final en el grafo, para poder generar explicaciones tenemos que adaptar este sistema a nuestro problema, para poder construir este grafo tenemos que tener claro:

- Una capa del grafo representa un tipo de argumento, ejemplo, preferencia de comida, tipo de cliente, etc.
- Un nodo dentro de una capa representa el valor del argumento, ejemplo, cliente familiar, comida italiana.

Este son basicamente las ideas del *framework* que presentan, el artículo tiene otra parte interesante en la cual quieren estudiar como los usuarios perciben el sistema de recomendación de manera diferente si se proporciona algún tipo de explicación con conocimiento en comparación con un sistema que no las proporcione.

Parten de la siguiente hipótesis: Un sistema que proporciona explicaciones tiene una diferencia significativa en la percepción del usuario y este tiene un impacto positivo en la experiencia de interacción de los usuarios y en su intención de usar el sistema con más frecuencia y en recomendarlo a otros.

La metodología usada fue una encuesta y los usuarios se distribuyeron en dos grupos aleatoriamente, a unos se les proporcionan explicaciones y a otros no.

Los resultados fueron los siguientes:

- El resultado principal fue que las explicaciones incrementaban significativamente la satisfacción percibida en el sistema de recomendación.
- Mostro que facilitar explicaciones no hace que el uso del sistema de recomendación sea más sencillo.
- La confianza en el sistema de recomendación no se ve aumentada por la explicación.

Este modelo en teoría se puede adaptar a cualquier tipo de problema, aunque su aplicación es mucho más sencilla si tenemos un recomendador basado en restricciones, como vimos en la sección 1.2.3. Otra técnica muy usada para recomendar es el razonamiento basado en casos, con lo cual para poder generar explicaciones en recomendadores que usen ese paradigma o que simplemente quieran generar las explicaciones basando en esta técnica, remitimos al lector a la sección 1.1.2 Explicaciones en razonamiento basado en casos para que vea como se generan y ejemplos concretos.

1.3.4 Comparativa de los modelos estudiados

A lo largo de estas subsecciones se ha realizado una revisión de las diferentes técnicas de generación de explicaciones en sistemas de recomendación, las principales características de estos modelos se encuentran resumidas en la Tabla 1.2

Tabla 1.2 Resumen de las principales características de los modelos vistos

Modelo	Características
(Herlocker, Konstand y Rield, 2000)	Se trata del estudio más completo que existe en la literatura sobre como poder generar explicaciones que justifiquen el proceso de recomendación en entornos colaborativos. No solo presenta explicaciones colaborativas sino que además presenta otras que están basadas en características de las películas.
(Vij, Sen y Rield, 2009)	Interesante trabajo que nos presenta un modelo para poder generar explicaciones basadas en las etiquetas que los usuarios ponen a las diferentes películas.
(Cramer <i>et al.</i> , 2008)	Presenta un modelo de explicaciones basado en las características de los ítems, los usuarios puntúan una serie de ítems y el sistema infiere automáticamente que características le gustan.
(Tintarev, 2010)	Presenta un modelo de explicaciones basado en el contenido, pero este modelo esta altamente personalizado a los gustos del usuario, las características que le gustan se le preguntan directamente no se infieren. También presenta otras explicaciones no tan personalizadas y más basadas en los ítems.
(Bilgic y Mooney, 2005)	Modelo de explicaciones basado en el contenido y colaborativas, utiliza palabras clave de los ítems para generar las explicaciones y las colaborativas siguen la misma idea de Herlocker
(Zanker y Ninaus, 2010)	Propone un framework para generar explicaciones basadas en conocimiento que puedan explicar el proceso de razonamiento seguido por el recomendador

En la siguiente Tabla 1.3 mostramos las ventajas y desventajas de cada uno de los modelos vistos.

Tabla 1.3 Ventajas y desventajas de los modelos presentados

Modelo	Ventajas	Desventajas
(Herlocker, Konstam y Riedl, 2000)	<ul style="list-style-type: none"> • Justifica proceso de razonamiento en enfoque colaborativo • Se usa en un entorno de películas como nuestro dominio • Intenta otro tipo de explicaciones para ver los efectos de la persuasión 	<ul style="list-style-type: none"> • No explica cómo se genera la recomendación, aunque esto es algo casi imposible para enfoques colaborativos • No queda del todo claro como se generan las explicaciones basadas en contenido • <i>MovieLens</i> si que se puede probar, pero estas explicaciones no están disponibles
(Vij, Sen y Riedl, 2009)	<ul style="list-style-type: none"> • Justifica proceso de razonamiento usando etiquetas • El sistema está disponible en <i>MovieLens</i> y se puede probar 	<ul style="list-style-type: none"> • Necesitas estar añadiendo etiquetas a las películas constantemente
(Cramer <i>et al.</i> , 2008)	<ul style="list-style-type: none"> • Genera explicaciones basadas en características de los ítems que el infiere automáticamente de las puntuaciones 	<ul style="list-style-type: none"> • No explica cómo se generan las recomendaciones • Recomienda obras de arte • Requiere que se voten obras para que el sistema pueda generar recomendaciones
(Tintarev, 2010)	<ul style="list-style-type: none"> • Genera explicaciones que están altamente personalizadas a los gustos del usuario • Usa también explicaciones que no estén personalizadas • Usa películas para hacer las pruebas • Con introducir los gustos el sistema ya es capaz de generar recomendaciones 	<ul style="list-style-type: none"> • No explica cómo se generan las recomendaciones • No se puede probar
(Bilgic y Mooney, 2005)	<ul style="list-style-type: none"> • Genera explicaciones basando en las palabras que tienen cada una de las características • También presenta explicaciones colaborativas para hacer 	<ul style="list-style-type: none"> • Recomienda libros • Requiere que se vote un número de libros determinados para que el sistema genere recomendaciones

	comparaciones	
(Zanker y Ninaus, 2010)	<ul style="list-style-type: none"> • Desacopla proceso de razonamiento de explicación • Puede usarse para cualquier proceso de recomendación usado 	<ul style="list-style-type: none"> • No proporciona la implementación del framework • Se usa en un dominio de balnearios

La siguiente Tabla 1.4 muestra los diferentes paradigmas que usan para generar la explicación.

Tabla 1.4 Tipos de explicaciones que generan los modelos vistos

Modelo	Colaborativo	Contenido	Conocimiento
(Herlocker, Konstam y Riedl, 2000)	Si	Si	No
(Vij, Sen y Riedl, 2009)	No	Si	No
(Cramer <i>et al.</i> , 2008)	No	Si	No
(Tintarev, 2010)	Si	Si	No
(Bilgic y Mooney, 2005)	Si	Si	No
(Zanker y Ninaus, 2010)	No	No	Si

Para finalizar la Tabla 1.5 expone los criterios por los que las explicaciones han sido evaluadas.

Tabla 1.5 Objetivos que analizan los modelos vistos

Modelo	Trans	Valid	Conf	Pers	Efec	Efic	Satis	Relev	Comp	Educ
(Herlocker, Konstand y Rield, 2000)				Si	Si	Si				Si
(Vij, Sen y Rield, 2009)	Si				Si					Si
(Cramer <i>et al.</i> , 2008)	Si		Si		Si		Si			Si
(Tintarev, 2010)				Si	Si					
(Bilgic y Mooney, 2005)					Si					
(Zanker y Ninaus, 2010)			Si		Si		Si			

Capítulo 2 – Modelos de conocimiento propuestos

En las secciones 1.2 y 1.3 vimos como funcionaban los diferentes tipos de paradigmas de recomendación y como se generaban sus explicaciones asociadas, haciendo una introducción en este dominio de los sistemas de recomendación, pero para poder generar un modelo explicaciones necesitamos hacer una representación más exhaustiva, capturando cual es su modelo de conocimiento. Analizaremos cual es el modelo de conocimiento de un sistema recomendación, este es un aspecto vital en cualquier sistema de conocimiento cuanto más completo y mas información sea capaz de capturar el modelo mayor será la capacidad de este para generar explicaciones.

Atendiendo al análisis que se hizo en la sección 1.1.1 de sistemas expertos, tratándolas de adaptar a nuestro dominio de sistemas de recomendación, podemos concluir que vamos a tener los siguientes tipos de explicaciones:

- *Terminológicas.* Estas explicaciones suministran la información semántica de los diferentes conceptos. Suelen responder a la pregunta de ¿Cuál es el significado de....?. Estas explicaciones son útiles en los recomendadores ya que pueden proporcionar información relativa a las propiedades de los ítems, por ejemplo, que es un megapixel en un recomendador de cámaras de fotos. También puede proporcionar información relativa a los requisitos que se le piden a un usuario, por ejemplo, en un recomendador basado en conocimiento en un contexto de ordenadores puede pedir un requisito que sea la cache, etc. Este tipo de explicaciones aunque no son de las más importantes sí que en determinados contextos pueden ser de utilidad.
- *¿Por qué?-Explicaciones.* Proporciona una justificación a un determinado evento, para un sistema de recomendación porque se me ha recomendado un cierto ítem. Suelen responder a la pregunta ¿Por qué el ítem.....es bueno para mí?, no hay que confundirlas con las de los sistemas de reglas, para nosotros tienen otro significado. En nuestro caso esta son las explicaciones más importantes y son las que tienen implementadas todos los modelos vistos, se trata de explicaciones que tratan de justificar porque un determinado ítem es apropiado para un usuario final.
- *¿Cómo?-Explicaciones.* Estas explicaciones describen el proceso que conduce a un determinado evento. En nuestro modelo estas explicaciones irán dirigidas al ingeniero del

conocimiento para que verifique diferentes aspectos del sistema. Casi ninguno de los modelos vistos los implementa ya que todos se centran sobre todo en los usuarios finales.

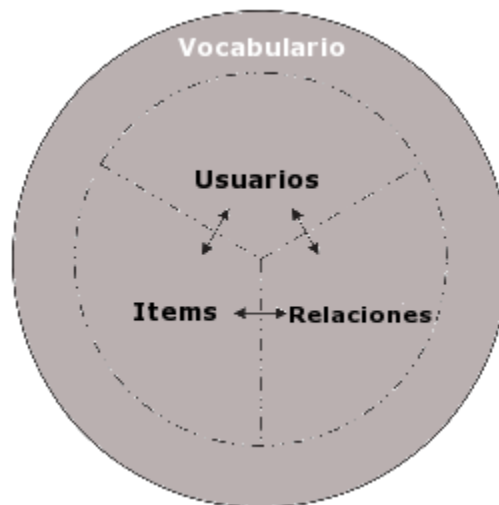
- *Explicaciones de propósito.* Este tipo de explicaciones tienen como objetivo describir el propósito de un hecho u objeto, suelen responder a la pregunta ¿Cuál es el propósito de....? o ¿Para qué es.....? Este tipo de explicaciones tienen sentido en recomendadores basados en conocimiento y en especial que sean de tipo conversacional, aquí el usuario puede preguntar cuál es el objetivo de proporcionar cierta información al sistema, como un cierto requisito o una puntuación.

Como vimos en la sección de explicaciones en CBR es necesario un buen modelo de conocimiento para generar el mayor número de explicaciones posibles.

2.1 Modelo de conocimiento de un sistema de recomendación

En esta sección seguiremos la idea de los contenedores de conocimiento de un sistema CBR, solo que en este caso lo adaptaremos a nuestro caso de los sistemas de recomendación, con lo que vimos en la sección 1.2 sobre el funcionamiento de los sistemas de recomendación ya somos capaces de identificar una serie de elementos clave, estos contenedores de conocimiento son cuatro, ver figura 2.1:

Figura 2.1 Los cuatro contenedores de conocimiento de un sistema de recomendación



- *Usuarios*, en este contenedor como su nombre indica se almacenara todos los datos básicos referentes a los usuarios, también será importante en el dominio de los sistemas de recomendación basados en conocimiento ya que en él se almacenaran los requisitos del usuario. A su vez en el sistema tendremos que identificar dos tipos de usuarios: Usuario Principal y Usuario Vecino, el primero será el usuario sobre el que se está realizando la recomendación y los segundos serán todos aquellos que estén relacionados con el principal. Hay que tener en cuenta aquí también, que habrá dominios como los sistemas de recomendación basada en contenidos en los que solo habrá un usuario y será el del principal.
- *Ítems*, este contenedor almacenara toda la información relativa a los ítems, en el dominio de sistemas de recomendación basados en contenido y conocimiento serán de importancia los datos de las propiedades del mismo. A su vez aquí distinguiremos dos tipos de ítems: Ítem Recomendado e Ítem Vecino, el primero será el que se le recomienda al usuario y el segundo serán todos aquellos que sean similares al recomendado. Al igual que pasaba con los usuarios vecinos, en las recomendaciones colaborativas puede que no tengamos ítems vecinos.
- *Relaciones*. Este contendrá todo el conocimiento relativo a las relaciones que existen entre los dos contenedores de conocimiento anteriores, las relaciones más importantes serán aquellas que nos permitan determinar la similitud entre ítems o usuarios y la relación que existe entre los requisitos de un usuario y las propiedades de un ítem. En el caso de relaciones de similitud al igual que hacíamos en la sección 1.1.2 con el CBR estas medidas pueden ser de dos tipos: Locales y globales.
- *Vocabulario*. Representa el elemento central del resto de contenedores de conocimiento, considera que información es relevante en ese modelo y como esa información puede ser expresada, definiendo los términos y estructuras del modelo del conocimiento. Como vimos en la sección 1.1.2 aunque este conocimiento no nos sea útil de una forma directa para poder definir el conocimiento usado para resolver el problema, si que nos será útil para responder cuestiones acerca del dominio. Existen muchas formas de capturar el modelo de conocimiento como por ejemplo los marcos y los objetos. Nosotros no vamos a

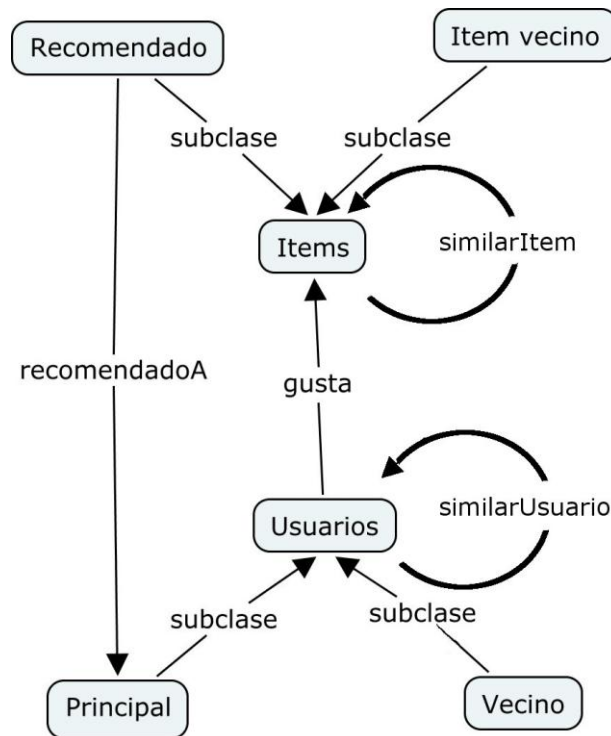
ser exhaustivos en hacer esta captura del conocimiento sino que nuestro objetivo será hacer una representación grafica del modelo de conocimiento a través de una red semántica en la cual podamos ver mejor cuales son las relaciones entre los diferentes conceptos.

Con lo cual cuanto más tengamos elaborado el modelo de conocimiento, mas poder explicativo tendrá nuestro sistema.

Todos los paradigmas de recomendación van a tener estos contenedores de conocimiento lo que les va a diferenciarlos es sobre todo el vocabulario y las relaciones que hay entre los diferentes contenedores, de forma que a continuación para cada paradigma de recomendación definiremos su modelo de conocimiento específico.

El primero que veremos será el modelo de conocimiento genérico de un sistema de recomendación colaborativo es el siguiente, ver figura 2.2:

Figura 2.2 Modelo de conocimiento de un sistema de recomendación colaborativo

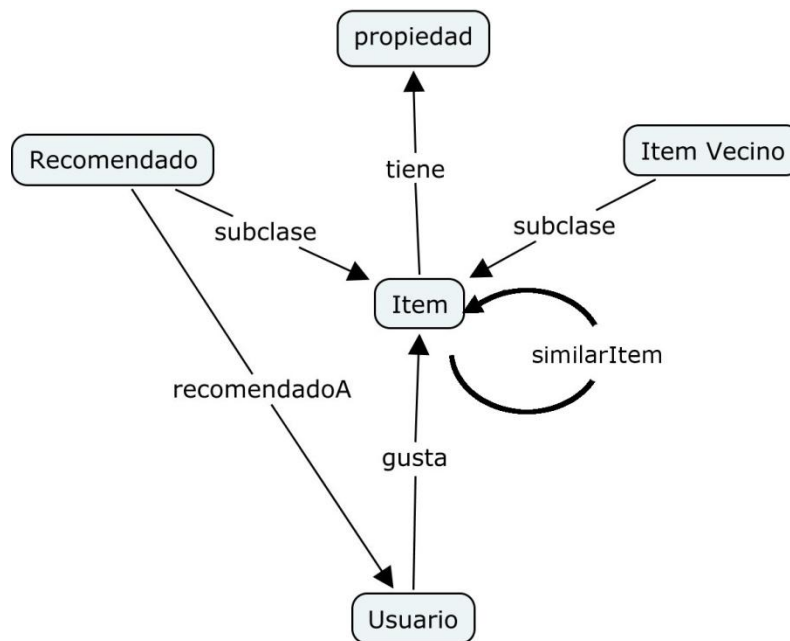


A partir de la imagen podemos identificar los siguientes contenedores:

- Usuarios, en este prototipo tendremos tanto usuarios vecinos como usuario principal.
- Ítems, también identificaremos dos tipo ítems, el que es recomendado y el vecino.
- Relaciones, identificamos las siguientes asociaciones entre los diferentes conceptos:
 - `similarUsuario(UsuarioA,UsuarioB)` establece una relación de similitud entre dos usuario del sistema.
 - `similarItem(Item1,Item2)` establece una relación de similitud entre dos ítems del sistema.
 - `recomendadoA(ItemRecomendado, UsuarioPrincipal)` establece una relación que quiere decir que el ítem es el que se le recomienda al usuario que está pidiendo una recomendación.
 - `gusta(Usuario, Item)` esta relación es la puntuación que le da un usuario a un ítem.

A continuación veremos cuál será el modelo de conocimiento de un sistema de recomendación basado en contenido, ver figura 2.3:

Figura 2.3 Modelo de conocimiento de un sistema de recomendación basado en contenido

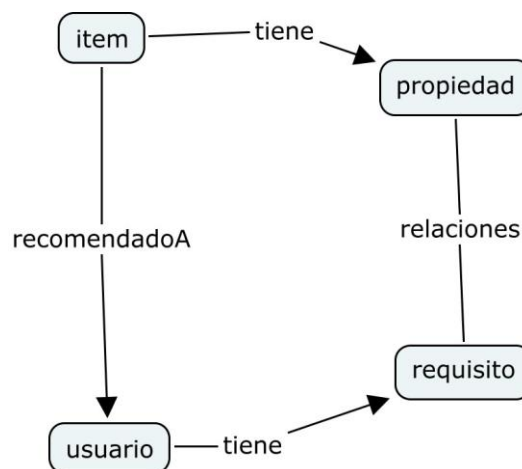


A partir de la imagen, podemos identificar los siguientes contenedores:

- Usuarios, en este modelo tan solo tendremos un usuario y será el usuario Principal.
- Items, en este modelo tendremos también dos tipos de ítems: Recomendado y Vecino, además los ítems almacenaran una serie de propiedades que los definan.
- Relaciones, identificaremos las siguientes asociaciones entre conceptos:
 - `similarItem(Item1,Item2)` se establece una relación de similitud entre dos ítems, en este tipo de sistemas esta similitud está relacionado con las propiedades que tienen los ítems.
 - `gusta(Usuario,Item)` esta relación es la puntuación que le da un usuario a un ítem.
 - `recomendadoA(ItemRecomendado, Usuario)` estable una relación que quiere decir que el ítem es el que se le recomienda al usuario que está pidiendo una recomendación.

El siguiente modelo que veremos será un sistema de recomendación basado en conocimiento, ver Figura 2.4:

Figura 2.4 Modelo de conocimiento de un sistema de recomendación basado en conocimiento



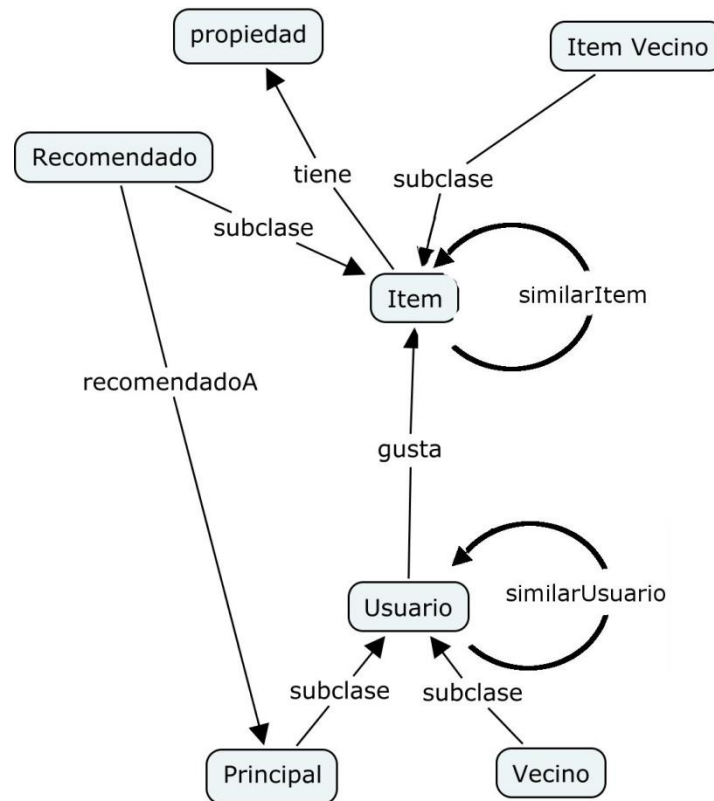
A partir de la imagen, podemos identificar los siguientes contenedores:

- Usuarios, en este caso solo tendremos al usuario principal y el aspecto importante es que almacenará los datos referentes a los requisitos sobre la recomendación que establece el usuario.
- Items, en este caso tan solo tendremos un ítem que será el recomendado, además este ítem almacenará los datos relacionados con las propiedades que tiene el mismo.
- Relaciones, este es el aspecto más importante de este tipos de modelos ya que es el que le da la capacidad de inferencia, el modelo de la Figura 2.4 es bastante genérico ya que todas estas relaciones son dependientes del sistema que estemos representando, en la sección 5.2 veremos un ejemplo completo.
 - recomendadoA(ItemRecomendado, Usuario) establece una relación que quiere decir que el ítem es el que se le recomienda al usuario que está pidiendo una recomendación.

Para finalizar el último modelo de conocimiento que veremos será para un sistema de recomendación híbrido, este esquema dependerá de la hibridación que realicemos, no es lo mismo una hibridación en cascada que una hibridación basada en combinación de características.

En cascada, por ejemplo, no habrá relaciones entre ambos dominios ya que las estrategias de recomendación que contenga son independientes. Mientras en combinación de características sí que habrá relaciones entre las características de ambos dominios y se basara en combinar los conceptos y relaciones de los modelos de las Figuras 2.2, 2.3 y 2.4 dependiendo de las estrategias a combinar, a continuación, mostraremos un ejemplo de cómo sería el modelo de conocimiento de un sistema de recomendación híbrido con combinación de características colaborativas y de contenido, ver Figura 2.5:

Figura 2.5 Modelo de conocimiento de un sistema de recomendación híbrido con combinación de características colaborativas y de contenido



A partir de la imagen, podemos identificar los siguientes contenedores:

- Usuarios, en este prototipo tendremos tanto usuarios vecinos como usuario principal.
- Ítems, en este modelo tendremos también dos tipos de ítems: Recomendado y Vecino, además los ítems almacenarán una serie de propiedades que los definan.
- Relaciones, identificamos las siguientes asociaciones entre los diferentes conceptos:
 - `similarUsuario(UsuarioA,UsuarioB)` establece una relación de similitud entre dos usuarios del sistema.
 - `similarItem(Item1,Item2)` establece una relación de similitud entre dos ítems del sistema, esta similitud puede ser tanto la referida a la relaciones de similitud referidas a una estrategia de recomendación colaborativa basada en ítems, como a las relaciones de similitud entre ítems de una estrategia basada en contenido.

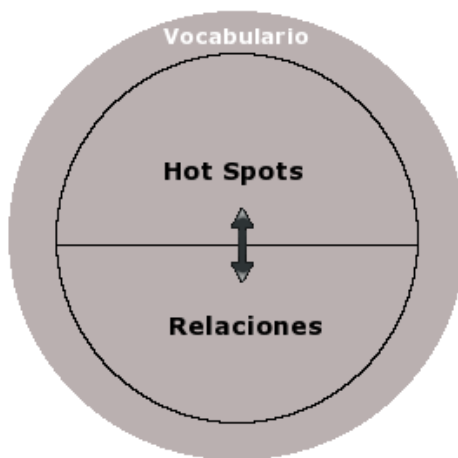
- recomendadoA(ItemRecomendado, UsuarioPrincipal) establece una relación que quiere decir que el ítem es el que se le recomienda al usuario que está pidiendo una recomendación.
- gusta(Usuario, Item) esta relación es la puntuación que le da un usuario a un ítem.

2.2 Modelo de conocimiento para el Ingeniero de conocimiento

Una característica de los sistemas que hemos visto es que ninguno ofrece explicaciones para el ingeniero del conocimiento que le permita poder verificar y validar el sistema, en nuestro modelo también nos centraremos en este tipo de usuarios, para ello generaremos explicaciones de tipo *¿Cómo?* que le dirán al ingeniero como ha sido el proceso de recomendación de un ítem.

Para poder generar estas explicaciones necesitaremos capturar las operaciones de procesamiento de información de nuestro sistema de recomendación que queramos validar o como nosotros lo hemos definido los “Puntos Calientes” de la aplicación, al igual que antes tendremos contenedores de conocimiento que son tres, ver figura 2.6:

Figura 2.6 Los tres contenedores de conocimiento en un sistema de recomendación usando un enfoque de procesos de información



- *Punto caliente (Hot Spots)*, es un contenedor que nos almacenara todo lo referente a la operación de procesamiento de información que queremos validar, este contenedor va almacenar tres categorías más de datos: Entradas que almacenaran

todo lo referente a la entrada de la operación de procesado, proceso todo lo referente al proceso que transforma las entradas en salidas y salidas lo referente a las salidas de la operación.

- *Relaciones*. Muchos de los puntos calientes van a estar relacionado entre sí, ya sea porque pertenezca al proceso de otro punto caliente o por relaciones de precedencia entre puntos calientes.
- *Vocabulario*, es el mismo concepto que en la subsección anterior, nos dará una visión global del modelo viendo como se relacionan entre si los puntos calientes.

Hemos optado por esta representación del conocimiento y no por la anterior ya que esta nos parece que reflejaba mejor el problema al que se enfrenta este tipo de usuarios, que es la verificación de las diferentes funciones del sistema. Con lo cual esta forma de representar le será mucho más intuitiva y cómoda.

A continuación vamos a dar una serie de pautas sobre como poder extraer los puntos calientes de la línea de razonamiento seguida por el recomendador:

1. Analizar el sistema de recomendación y clasificarlo según las técnicas vistas en el apartado 1.2 Tipos de sistemas de recomendación.
2. Del paso anterior el sistema habrá quedado dividido en los siguientes módulos:
 - Hibridación en paralelo: Mixtos, ponderados y conmutación
 - Hibridación monolítica: Combinación de características y aumento de características.
 - Pipeline de hibridación: Cascada y meta-nivel.
 - Técnicas básicas: Colaborativas, basadas en contenido y basadas en conocimiento.

Cada uno de estos módulos ya son puntos calientes que podemos seleccionar, si alguno de los módulos es a su vez alguna técnica de hibridación se seguirán descomponiendo en más técnicas que pueden seguir siendo hibridas o por el contrario pueden ser una técnica básica.

3. Como hemos indicado anteriormente cada una de estas estrategias es un potencial punto caliente, si lo seleccionamos tendremos que identificar cuáles son sus entradas, salidas y el proceso, al analizar el proceso podemos a su vez identificar más puntos calientes y seguir descomponiéndolo.

Un ejemplo completo de este tipo de análisis del modelo de conocimiento lo podéis ver en la sección 5.3 de la memoria.

2.3 Modelo de conocimiento de la explicación

Una vez tenemos representado el modelo de conocimiento de nuestro sistema de recomendación ya sea siguiendo el esquema de representación de la sección 2.1 o 2.2, ahora toca definir el modelo de conocimiento de la explicación, el modelo de conocimiento de las explicaciones van a seguir un esquema parecido al modelo de conocimiento para el ingeniero de conocimiento, en este caso las explicaciones van a ser también unidades de procesamiento de información, la principal característica es que como datos de entrada van a tomar piezas de información generadas en los modelos anteriores, en este caso tendremos un único contenedor de conocimiento:

- Explicación, por cada explicación que queramos generar en el sistema tendremos que tener uno, este a su vez estará descompuesto en los siguientes tipos de datos:
 - Entrada, respecto a los modelo de conocimiento del sistema de recomendación identificaremos que datos serán los que vayan a ser utilizados para generar esta explicación.
 - Proceso, aquí no definiremos nada, pero es básicamente la transformación que se realiza a los datos de entrada para que se genere la interfaz de salida.
 - Salida, aquí tan solo especificaremos si la salida es una interfaz grafica o textual.

Al igual que con el modelo de conocimiento, en la secciones 5.2 y 5.3 de la memoria tendremos ejemplo de este tipo de análisis del modelo de conocimiento de las explicaciones.

Capítulo 3 – Aspectos de diseño

En la sección anterior hemos visto como poder representar el conocimiento que será necesario para poder generar la explicación, pero tanto la especificación de ese conocimiento como la posterior forma que tenga la explicación final dependerá enormemente de los siguientes aspectos: objetivos, audiencia, foco, extracción de la información y la forma en la que la representamos. A lo largo de esta sección describiremos detalladamente cada uno de estos aspectos y como van a influir en la explicación final.

Los objetivos es uno de los principales aspectos que hay que tener en cuenta a la hora de diseñar las explicaciones, cual es el propósito de generar explicaciones en un sistema de recomendación, como vimos en la sección 1.3.1 tenemos diez objetivos clave:

- *Transparencia.* Cómo funciona el sistema.
- *Validez.* Permite a los usuarios ver que está mal en el razonamiento del sistema.
- *Confianza.* Incrementar la confianza del usuario en el sistema.
- *Persuasión.* Convencer a los usuarios para probar o comprar.
- *Efectividad.* Ayuda al usuario a tomar mejores decisiones.
- *Eficiencia.* Ayuda al usuario a tomar decisiones más rápido.
- *Satisfacción.* Incrementa la facilidad de uso o de disfrute.
- *Relevancia.* Justifican por que determina información se necesita del usuario.
- *Comprensibilidad.* Relacionar los conceptos conocidos del usuario con los conceptos empleados por el recomendador.
- *Educación.* Ayudan al usuario a que entiendan mejor el dominio del producto.

El contenido y objetivo de la explicación varía en función de a quien vaya dirigida la misma, en este contexto vamos a distinguir los siguientes tipos de usuarios:

- *Ingeniero del conocimiento.* Este tipo de usuario cuando hace uso del sistema este interesado en las operaciones que realiza el sistema de recomendación. Con lo cual una explicación dirigida a este tipo de usuarios va a tener como objetivo principal la transparencia y la validez, porque su interés radica en observar las operaciones internas que haya realizado el sistema de recomendación. En cuanto a la especificación del modelo de conocimiento van a usar la que describimos en la sección 2.2.

- *Usuarios finales.* Estos usuarios por líneas generales van a estar interesados en una mejor comprensión de la recomendación hecha por el sistema o una justificación de porque deberían comprar o probar un producto recomendado. En este tipo de usuarios sí que podemos tener en cuenta mas objetivos, por ejemplo, podemos generar explicaciones que persuadan al usuario a comprar nuestros productos o por el contrario usar una transparencia absoluta para que el usuario confié mucho más en nuestro sistema, etc. En lo referente a la especificación de su modelo de conocimiento al no centrarse tanto en el proceso usaremos los modelos definidos en la sección 2.1.

El foco de la explicación es un aspecto que juega un rol importante en los requisitos y forma que va tener la explicación. Por lo general, tenemos dos tipos de focos:

- *Proceso.* Una explicación que ponga el foco en el proceso se centra en presentar la información relacionada con el proceso de recomendación. Este tipo de explicaciones responderá a las cuestiones de cómo se ha generado una determinada recomendación, estas son de mayor interés para el ingeniero del conocimiento.
- *Solución.* Una explicación que se centre en este aspecto, se preocupa de la salida del proceso de recomendación, en nuestro caso un ítem recomendado. Estas explicaciones se refieren comúnmente como una justificación de porque se ha recomendado un determinado ítem. Una explicación en este contexto va consistir en un argumento a favor de la recomendación.

Un elemento clave en la construcción de la explicación es como extraemos el conocimiento usado en la fase de resolución del problema por parte del recomendador y lo unimos a la explicación. Existen dos maneras de hacerlo:

- *Introspección.* La explicación utiliza un registro en donde se ha ido almacenando información referente al proceso de resolución usado por el recomendador y también información contenida en la consulta hecha por el usuario.
- *Independiente.* Construye la explicación que justifica una recomendación sin tener en cuenta como esa recomendación fue hecha, esto es, no tiene en cuenta el proceso de razonamiento seguido por el recomendador.

También se pueden combinar ambas formas dependiendo de nuestros objetivos y el tipo de usuario al que estemos generando las explicaciones, por ejemplo, un ingeniero del conocimiento esta mucho más interesado en cómo funciona el sistema requerirá que la explicación haya usado conocimiento procedente de la introspección del sistema.

En cuanto a la introspección del sistema, la información nos vendrá en forma de trazas que contendrán datos clave usados durante la fase de recomendación del sistema y que le llevaron a realizar una conclusión: el valor de variables, cálculos, etc. El desarrollador es el que se encarga de seleccionar los puntos del programa en los que quiere extraer información que le sea de utilidad para construir la explicación.

Todas las características descritas anteriormente producen toda la información conceptualmente, ahora solo falta presentar de una forma eficiente toda esa información al usuario. Los dos métodos que disponemos son:

- *Lenguaje Natural*. Esta forma se realiza mediante mensajes de texto generados de forma automáticamente, mediante la técnica de encaje de patrones.
- *Forma grafica*. Aquí existen muchas formas de representar la información, por ejemplo, mediante histogramas como veíamos en la sección 1.3.2. Como consejo se tienen que elegir figuras que sean simples y directas, ya que demasiada información y representada de una forma abstracta puede incluso producir resultado contraproducentes.

Capítulo 4 – Modelo de explicación propuesto

4.1 Introducción

En este capítulo presentaremos y describiremos el modelo que hemos realizado, nuestro modelo está orientado a la creación de explicaciones cuyos objetivos principales sean los de aumentar la transparencia, validez y confianza en el sistema de recomendación, esto es debido a que el modelo de explicaciones propuesto tratará de reconstruir la línea de razonamiento seguida por el sistema al realizar el proceso de recomendación, esto lo conseguimos a través de la introspección del mismo.

Otro aspecto importante de este modelo es que las explicaciones se construirán con retrospectiva, esto quiere decir, primero se ha generado la recomendación con lo cual su resultado ya se conoce y después en función de si el usuario del sistema lo desea puede pedir una explicación, si quiere, entonces se le reconstruye la explicación en función de los datos que haya recogido en el proceso de razonamiento. Este tipo de planteamiento nos permite desacoplar el proceso de razonamiento del de generación de la explicación, esto tiene sus ventajas:

- Evita tener que reducir la competencia del sistema a favor de una mayor comprensibilidad.
- El sistema se vuelve más escalable al tener separados los dos conceptos.

Pero también sus desventajas:

- Se requiere un conocimiento adicional, un proceso diferente para construir la explicación y un coste adicional por tener que mantener también un sistema de explicaciones.

En cuanto al foco de nuestras explicaciones está claro que es el proceso que se ha usado para generar la recomendación.

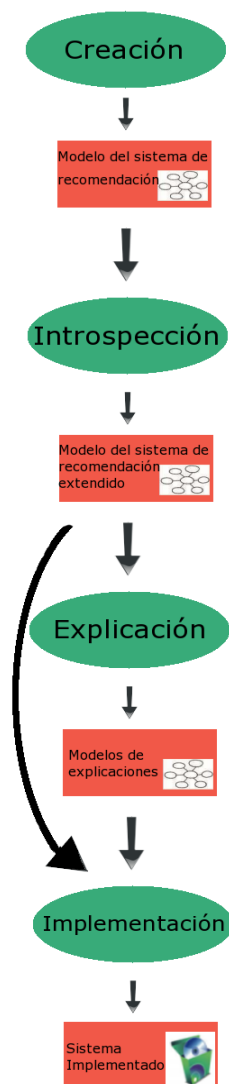
El modelo que proponemos está creado tanto para atender a las necesidades de explicación de un ingeniero del conocimiento como para un usuario final y las interfaces que te permite generar son tanto textuales como gráficas.

4.2 HotSpot Explanations: Un modelo para generar explicaciones

En esta sección presentamos el modelo que hemos desarrollado, este está dividido en dos fases, una primera fase de diseño del sistema, ver Figura 4.1, y otra fase de ejecución que es referente a la ejecución del sistema que hemos desarrollado en la fase de diseño, ver Figura 4.2.

Como se indica en la Figura 4.1, la fase de diseño del modelo tiene las siguientes actividades:

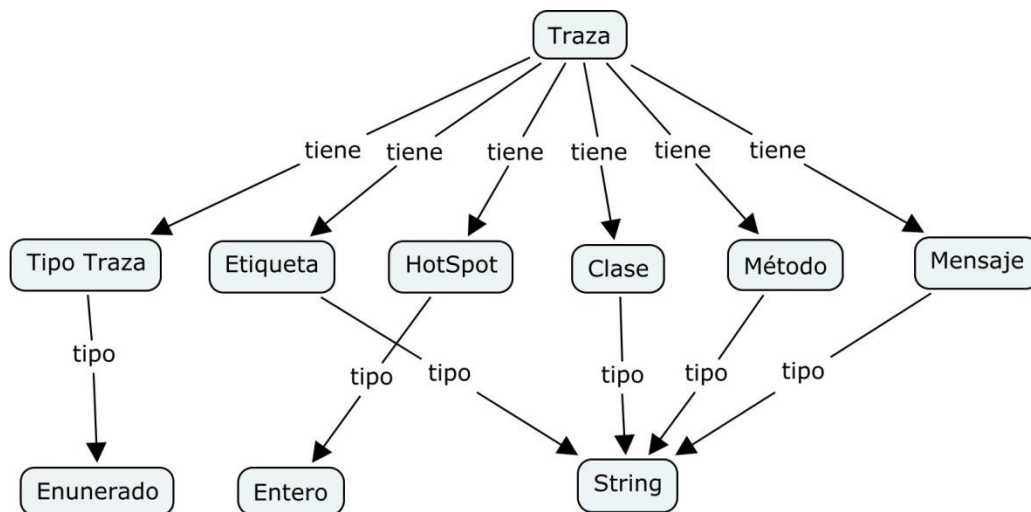
Figura 4.1 Fase de diseño del sistema HotSpots Explanations



- *Creación.* En esta primera fase se creara el modelo de dominio necesario para poder generar la explicación basándose en el funcionamiento del sistema de recomendación, por lo tanto este modelo será una descripción del sistema de recomendación. Para elaborarlo utilizamos los modelos de conocimiento que hemos desarrollado en el capítulo 2 en las secciones 2.1 y 2.2 dependiendo del usuario al que queramos diseñar las explicaciones.
- *Introspección.* Con el modelo que hemos generado en el paso anterior, tendremos todos los conceptos y relaciones definidas, ahora toca para cada uno de esos datos tendremos que identificar en qué punto del sistema de recomendación implementado se extraen, una vez identificados generaremos la traza asociada a ese punto del programa. A su vez cada una de estas trazas se asociaran a los conceptos y relaciones definidos en el modelo del sistema de recomendación y se nos generara un modelo del sistema de recomendación extendido, que incluirá como antes la descripción del funcionamiento del sistema pero ahora tendrá también cada uno de los datos identificados con la traza que identifica en qué punto se extrae.

En cuanto a la forma que tienen las trazas, siguen la estructura definida por (Carboneras, Álvarez, Martínez, 2012) y descrita en la Figura 4.2, donde:

Figura 4.2 Estructura de una traza



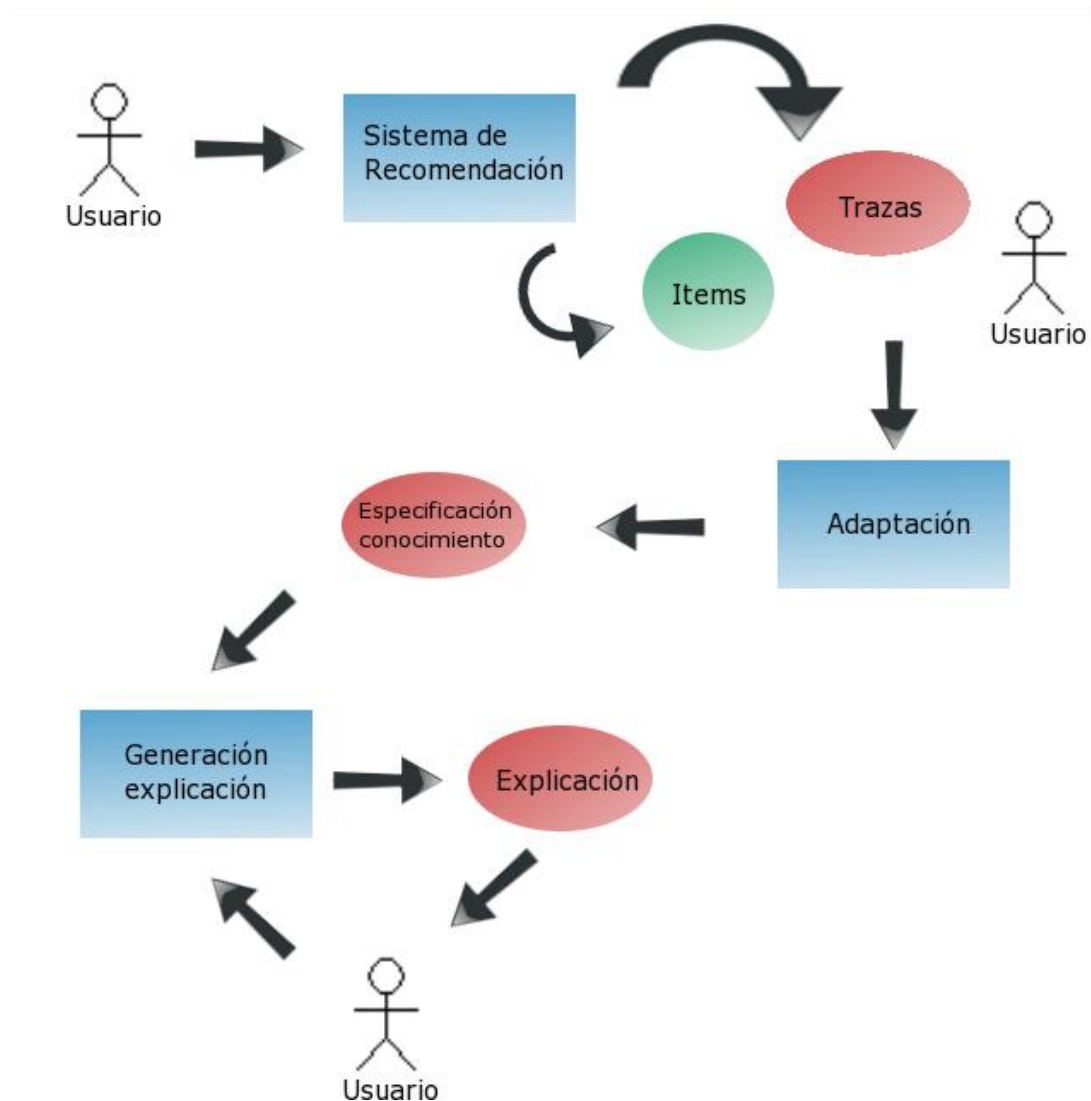
- *Tipo traza*, especifica si la traza es de tipo: Comentario, cálculo o una variable.
- *Etiqueta*, Si la traza se refiere a una variable del sistema de recomendación me dice el nombre identificativo de la variable (vacío si es de tipo cálculo o comentario).
- *HotSpot*, Punto caliente al que pertenece la variable (vacío si es de tipo cálculo o comentario).
- *Clase*, clase del sistema de recomendación que genera el mensaje.
- *Método*, método de la clase del sistema de recomendación que genera el mensaje.
- *Mensaje*, mensaje a mostrar, ya sea un comentario, un cálculo entero, el valor de una variable asociada, etc.

Como vemos con esta información ya somos capaces tanto de detectar donde está el dato en el sistema de recomendación como la información que va a contener para luego poder adaptarla al modelo del sistema de recomendación.

- *Explicación*. En esta actividad generaremos los modelos relacionados con las explicaciones, este modelo será una descripción de cómo serán las explicaciones usando el conocimiento del modelo generado en el paso anterior. Este modelo utilizará el modelo de conocimiento descrito en el capítulo 2 en la sección 2.3.
- *Implementación*. En esta etapa generaremos el programa que nos permita realizar las explicaciones, para ello usaremos los modelos que hemos generado en las etapas anteriores y los implementaremos para poder obtener el programa asociado. Las partes que tienen un mayor coste serán las referentes a la adaptación de las trazas a los diferentes conceptos, sobre todo porque muchas veces estas trazas tendrán un proceso de parseo del mensaje de la traza. El otro proceso que tendrá un mayor coste será el de generación de la explicación sobre todo la parte de definir el proceso ya que esta como vimos en la sección 2.3 no lo definimos en el modelo de conocimiento. El paso del modelo conceptual como los que definimos en el capítulo 2 Modelos de conocimiento propuestos a la codificación de un lenguaje de objetos como Java es prácticamente automática será la parte que tenga menor coste.

En la figura 4.3 se ve la fase de ejecución del sistema HotSpots Explanations una vez implementado:

Figura 4.3 Fase de ejecución del sistema HotSpots Explanations



El proceso que se llama Sistema de Recomendación es el sistema de recomendación que ya teníamos implementado inicialmente y del que queremos generar las explicaciones.

A continuación mostramos todo lo referido al sistema HotSpot Explanations:

- Las entradas del sistema HotSpot Explanations son las siguientes:
 - *Trazas*, son los datos que identificamos en la actividad de introspección de la fase de diseño. Cuando el usuario pide una recomendación este le

responderá con los ítems recomendados pero también internamente generará un XML con todas las trazas.

- *Ítem recomendado*, como he indicado anteriormente el usuario pedirá una recomendación al sistema de recomendación y este le responderá, con los ítem y trazas.
- *Usuario*, al inicio de la Figura 4.3 se ve como el usuario pide una recomendación y el sistema le responde con la recomendación, después el usuario puede pedir una explicación sobre un ítem en particular, en este punto es importante el usuario porque dependiendo del tipo que sea, ingeniero del conocimiento o usuario final, podrá exigir una explicación de tipo ¿Por qué? o ¿Cómo?, con lo cual tanto la especificación del conocimiento como las posteriores explicaciones cambiarán mucho.
- *Especificación del conocimiento*, esta especificación será una instancia del modelo del sistema de recomendación que describimos en la fase de diseño, será un modelo que solo contendrá los datos relacionados con el ítem recomendado.
- *Explicación*, este será una de las instancias de los modelos de explicación que definimos en la fase de diseño.
- Los procesos, tenemos dos procesos principales que se encargaran de reconstruir la línea de razonamiento del sistema de recomendación y de generar una explicación asociada:
 - *Adaptación*, este proceso será el encargado de realizar la adaptación de las trazas que recibe como parámetro de entrada HotSpot Explanations a la especificación del conocimiento en función del ítem recomendado y del tipo de usuario.
 - *Generación explicación*, este proceso será el encargado de generar a partir de la especificación del conocimiento uno de los modelos de explicación que especificamos en la fase de diseño, esta explicación dependerá del tipo de usuario y también del nivel de detalle que defina, como se aprecia en la Figura 4.3 al final el usuario puede volver a pedir una explicación con un mayor nivel de detalle sobre la explicación que se le haya mostrado.

Capítulo 5 – Implementación

En este capítulo implementaremos nuestro modelo de explicaciones en un caso real, en la primera parte describiremos el sistema de recomendación que hemos elegido para el caso de estudio, Moodvie, y cuáles son sus principales propiedades y características. En la segunda parte del capítulo describiremos cuales son las diferentes explicaciones que hemos desarrollado para el sistema.

5.1 Caso de estudio Moodvie

El sistema de recomendación elegido para implementar el modelo ha sido Moodvie (Garrido, Peña, Salmerón, 2011), se trata de un recomendador de películas que las filtra basándose en la teoría de los temperamentos de (Keirsey, 2002) y en el estado de ánimo del usuario. En la Figura 5.1 podemos ver un esquema conceptual de las diferentes estrategias que usa para generar la recomendación.

Figura 5.1. Esquema conceptual de Moodvie usando la taxonomía de sistemas de recomendación descrita en estado del arte



El primer paso será describir las fuentes de conocimiento que usa Moodvie para recomendar:

- *Perfil de usuario.* Este perfil de usuario básicamente almacenara el temperamento del usuario y estado de ánimo, los temperamentos son básicamente cuatro y son descrito según (Keirsey, 2002) de la siguiente manera:
 - *Artesano.* La habilidad natural de destacar en algún arte, desde las bellas artes hasta el arte de la dialéctica o los negocios. Amantes de la diversión, realistas pero optimistas, disfrutan del presente. Creativos y espontáneos, en busca de causar impacto en los demás. Confían en sus impulsos, creen que todo les saldrá bien.
 - *Guardián.* Piedra angular de la sociedad, personas entregadas a servir y preservar las instituciones. Gran capacidad de gestión de bienes y servicios. Buenos supervisores. Personas leales, obedientes, responsables y muy trabajadoras. Actúan con cautela, buscan seguridad en sus vidas. Adoran la justicia y la pertenencia a un grupo.
 - *Idealista.* Pasionales y entregados al crecimiento personal. Se esfuerzan en descubrir su verdadero yo, en mejorar como personas. Disfrutan ayudando a la gente y creando vínculos. Anhelan una vida romántica y la sabiduría. Intensos y espirituales. Gente muy auténtica, querida y ética en sus acciones.
 - *Racional.* Resolutores de problemas. Analizan el mundo con intención de entender cómo funcionan las cosas. Escépticos y pragmáticos. Orgullosos de su ingenio. Líderes estratégicos, con fuerza de voluntad. Tranquilos y serenos, se dejan guiar por la lógica. Disfrutan con la tecnología. A veces se ven como personas frías y distantes.

Para poder relacionar esta teoría con el usuario, cuando se registra en el sistema se le hace rellenar un cuestionario para poder saber su temperamento, este cuestionario esta descrito también en (Keirsey, 2002).

Al final quedara grabado en el perfil para cada uno de los temperamentos descritos anteriormente un valor que reflejara cual su porcentaje.

Por último, el estado de ánimo, este se obtiene preguntándole de manera explícita al usuario como quiere sentirse pudiendo elegir las siguientes opciones: sin estado de ánimo, esto es, que le recomiende basándose únicamente en su temperamento o con estado de ánimo, aquí puede seleccionar los siguientes estados: Sorprendido, alegre, relajado, triste y asustado.

- *Películas.* Estos son los elementos que se recomiendan al usuario, en cuanto a las propiedades de las películas vamos almacenar las siguientes características: Título, género principal, duración, nacionalidad, fecha de estreno, director, reparto, productora, calificación edades, guionista, distribuidora, sinopsis y distribución de géneros. A su vez vamos a tener once géneros diferentes: Acción, animación, aventuras, comedia, drama, fantasía, romántica, terror, thriller y ciencia ficción.

Todas las propiedades menos la distribución de géneros se extraen de la página web de La Guía del Ocio (<http://www.guiadelocio.com/>). La extracción de la distribución de géneros es un proceso más complicado que se resume en los siguientes pasos:

1. Crear un conjunto de casos base resuelto manualmente, aquí buscamos películas que sean representantes de cada uno de los géneros, de forma que para cada uno de los once géneros descritos anteriormente se buscan representantes que sean lo más puro posible. Para cada uno de estas películas almacenamos su sinopsis y su distribución de géneros.
 2. Para cada película que queremos categorizar usamos la técnica de CBR textual comparando la sinopsis de la película que queremos categorizar con la base de casos, esto junto al método KNN determinamos cuáles son las películas más parecidas. Una vez que sabemos cuáles son los vecinos más próximos como sabemos sus distribuciones de géneros usamos una serie de formulas que nos permitirán saber cuál es la distribución de géneros de la nueva película.
- *Tablas de conocimiento.* Estas tablas han sido realizadas por los autores y permiten relacionar los temperamentos con los géneros y por otro lado obtener

cual es el peso de cada estrategia de recomendación en la etapa de ponderación, las tablas que relacionan los temperamentos y géneros son tres:

- Tabla temperamento/genero cines, esta tabla simula la distribución de temperamentos en cada uno de los géneros, por ejemplo, en una película que fuera 100% acción los temperamentos se distribuirán de la siguiente manera 45,00% Artesano, 21,67% Guardián, 21,67% Idealista y 11,67% Racional. El resultado es la Tabla 5.1.

Tabla 5.1 Tabla temperamento/genero cines

	Acción	Animación	Aventuras
Artesano	45,00%	31,67%	40,00%
Guardián	21,67%	13,33%	20,00%
Idealista	21,67%	30,00%	26,67%
Racional	11,67%	25,00%	13,33%
	Comedia	Documental	Drama
Artesano	25,00%	11,67%	10,00%
Guardián	25,00%	36,67%	31,67%
Idealista	31,67%	20,00%	35,00%
Racional	18,33%	31,67%	23,33%
	Fantasia	Romántica	Terror
Artesano	30,00%	13,33%	31,67%
Guardián	22,67%	28,33%	24,00%
Idealista	30,00%	40,00%	24,33%
Racional	17,33%	18,33%	20,00%
	Thriller	Ciencia ficción	
Artesano	13,33%	10,00%	
Guardián	38,33%	21,67%	
Idealista	28,33%	28,33%	
Racional	20,00%	40,00%	

- Tabla temperamento/genero gustos, esta tabla lo que hará será para cada género y temperamento darle una puntuación que refleje cuanto le gusta a ese temperamento particular ese género también particular. Ver Tabla 5.2.

Tabla 5.2 Tabla temperamento/genero gustos

Acción		Animación	Aventuras
Artesano	8	7,6666667	10
Guardián	6,6666667	6	5,6666667
Idealista	5,3333333	7,3333333	6,6666667
Racional	3	7	6,3333333
Comedia		Documental	Drama
Artesano	6,3333333	3,3333333	3,6666667
Guardián	7	8	9,3333333
Idealista	8	7,6666667	8
Racional	6,6666667	9,3333333	7,3333333
Fantasía		Romántica	Terror
Artesano	7,6666667	5	7
Guardián	7	6,6666667	5
Idealista	7	10	2,6666667
Racional	6,6666667	6	4,6666667
Thriller		Ciencia ficción	
Artesano	6,6666667	7,3333333	
Guardián	7,3333333	4,3333333	
Idealista	4,6666667	4,6666667	
Racional	7,6666667	9,3333333	

Después todas las puntuaciones que ha dado un mismo temperamento a los géneros se normalizan de forma que nos van a quedar valores de 0 a 100. Ver Tabla 5.3.

Tabla 5.3 Tabla temperamentos/genero gustos normalizada

Acción		Animación	Aventuras
Artesano	11,01%	10,55%	13,76%
Guardián	9,13%	8,22%	7,76%
Idealista	7,41%	10,19%	9,26%
Racional	4,05%	9,46%	8,56%
Comedia		Documental	Drama
Artesano	8,72%	4,59%	5,05%
Guardián	9,59%	10,96%	12,79%
Idealista	11,11%	10,65%	11,11%
Racional	9,01%	12,61%	9,91%
Fantasía		Romántica	Terror
Artesano	10,55%	6,88%	9,63%
Guardián	9,59%	9,13%	6,85%
Idealista	9,72%	13,89%	3,70%
Racional	9,01%	8,11%	6,31%
Thriller		Ciencia ficción	
Artesano	9,17%	10,09%	
Guardián	10,05%	5,94%	
Idealista	6,48%	6,48%	
Racional	10,36%	12,61%	

- Tabla temperamento/genero combinada, debido a las ineficiencias que presentaban las dos anteriores tablas a la hora de recomendar, los autores

deciden combinar esas dos tablas para obtener mejores resultados, esta unión la realizan mediante la siguiente fórmula:

$$tablaCombinada(t, g) = \frac{tablaCines(t, g) * tablaGustos(t, g)}{\sum_{g2=Acción}^{Ciencia Ficción} tablaCines(t, g2) * tablaGustos(t, g2)}$$

Donde t es el temperamento g es el género de la posición de la tabla

Para las formulas que veremos en los apartados siguientes esta tabla será conocida como la *tabla3*. En resultado final se muestra en la Tabla 5.4.

Tabla 5.4 Tabla temperamento/genero combinada

	Acción	Animación	Aventuras
Artesano	18,78%	12,66%	20,86%
Guardián	7,43%	4,11%	5,83%
Idealista	5,40%	10,28%	8,31%
Racional	2,02%	10,12%	4,88%
	Comedia	Documental	Drama
Artesano	8,26%	2,03%	1,91%
Guardián	9,00%	15,09%	15,20%
Idealista	11,84%	7,17%	13,09%
Racional	7,07%	17,10%	9,90%
	Fantasia	Romántica	Terror
Artesano	12,00%	3,48%	11,56%
Guardián	8,16%	9,72%	6,17%
Idealista	9,82%	18,70%	3,03%
Racional	6,68%	6,36%	5,40%
	Thriller	Ciencia ficción	
Artesano	4,64%	3,82%	
Guardián	14,46%	4,83%	
Idealista	6,18%	6,18%	
Racional	8,87%	21,59%	

A continuación vemos la última tabla que es la tabla de reparto de recomendadores en la cual cada temperamento me va decir cual es peso de cada una de las estrategias de recomendación en el resultado de la puntuación final. El resultado está en la Tabla 5.5.

Tabla 5.5 Tabla reparto pesos recomendadores

	Peso Mixto	Peso Genero	Peso Temperamento
Artesano	35%	31%	34%
Guardián	32%	39%	29%
Idealista	36%	28%	36%
Racional	38%	34%	28%

El siguiente paso será describir la estrategia de recomendación, de acuerdo con el esquema de la Figura 5.1, Moodvie sigue las siguientes estrategias de recomendación:

- *Estrategia sin estado de ánimo.* Esta estrategia está basada en conocimiento, aquí como el usuario quiere obtener una recomendación que solo se base en el temperamento, aquí no se filtrara ninguna película todas ellas pasaran a la siguiente etapa de ponderación.
- *Estrategia con estado de ánimo.* También basada en conocimiento, en esta el usuario especifica cómo quiere sentirse después de ver la película en un estado de ánimo, para ello se filtran las películas que pasan a la siguiente etapa siguiendo el criterio siguiente:
 - Si quiere sentirse sorprendido, las películas que pasen a la siguiente etapa del recomendador serán aquellas que sobrepasen el X% de acción o el X% de aventuras.
 - Si quiere sentirse alegre, aquellas películas que sobrepasen el X% de comedia.
 - Si quiere sentirse asustado, aquellas películas que sobrepasen el X% de terror o el X% de thriller, siempre y cuando la animación este por debajo del X%.
 - Si quiere sentirse triste, aquellas películas que sobrepasen el X% de drama.
 - Si quiere sentirse relajado o tranquilo, aquellas películas que sobrepasen el X% de romántica o el X% de documental.

X será una constante que especifique el diseñador del sistema.

- *Proceso Conmutación.* El proceso de conmutación elegirá la estrategia que el usuario ha elegido cuando ha hecho la solicitud de recomendación, como vimos en el punto de creación de perfil de usuario antes de realizar la recomendación el sistema le pregunta al usuario que si quiere una recomendación con estado de ánimo o sin él, si elige con ánimo entonces se le pide que seleccione como quiere sentirse. Con lo cual en función de lo que especifique en ese punto a si seleccionara una estrategia u otra.
- *Estrategia mixta.* Esta estrategia sigue la idea más intuitiva de todas, consiste en combinar la tabla temperamento/genero combinada (*tabla3*) junto al temperamento del usuario y la película a recomendar. Por lo tanto para obtener la puntuación de una película se usa la siguiente fórmula:

$$puntuacion(u,p) = \sum_{g=Accion}^{Ciencia Ficción} \sum_{t=Artesano}^{Racional} Usuario(u,t) * Pelicula(p,g) * tabla3(t,g)$$

Donde:

$puntuacion(u,p)$ es la puntuación que obtiene un usuario u para una película p .

$Usuario(u,t)$ es una función que me devuelve el porcentaje de temperamento del usuario.

$Pelicula(p,g)$ es una función que me devuelve el porcentaje de género de la película.

$tabla3(t,g)$ me devuelve el valor que haya en la tabla de temperamento/genero combinada.

- *Estrategia de género.* La idea de esta estrategia será la de transformar los temperamentos del usuario en una distribución de géneros equivalente para después calcular la similitud que existe con la distribución de géneros de la película. Para extraer el género de un temperamento del usuario usamos la siguiente fórmula:

$$Usuario(u,g) = \sum_{t=Artesano}^{Racional} Usuario(u,t) * tablas3(t,g)$$

La similitud entre un género del usuario y la película se determina mediante la siguiente fórmula:

$$similitud(u, p, g) = 1 - \frac{|Usuario(g) - Pelicula(p, g)|}{100}$$

Con todo esto el último paso es determinar la puntuación de la película, para ello usamos la siguiente fórmula:

$$puntuacion(u, p) = \frac{\sum_{g=Accion}^{Ciencia Ficción} similitud(u, p, g)}{|Generos|}$$

- *Estrategia de temperamento.* Esta estrategia tiene la estrategia contraria a la anterior, consiste en transformar los géneros de la película en una distribución de temperamentos equivalente y luego determinar cuál es su similitud respecto a la distribución de temperamentos del usuario. Para extraer el temperamento de un genero de la película usamos la siguiente fórmula:

$$Pelicula(p, t) = \sum_{g=Accion}^{Ciencia Ficción} Pelicula(p, g) * tablas3(t, g)$$

La similitud entre un temperamento de la película y del usuario de la siguiente forma:

$$similitud(u, p, t) = 1 - \frac{|Usuario(u, t) - Pelicula(p, t)|}{100}$$

La puntuación de la película se calculara mediante la siguiente fórmula:

$$puntuacion(u, p) = \frac{\sum_{t=Artesano}^{Racional} similitud(u, p, t)}{|Temperamentos|}$$

- *Proceso de ponderación.* El proceso de ponderación combinara las puntuaciones de las estrategias mixta, genero y temperamento para devolver una puntuación final, para esto lo primero es determinar cuál es el peso que da el sistema a cada una de las estrategia de recomendación en función del temperamento del usuario y de la tabla de reparto de recomendadores que vimos anteriormente, para ello usa la siguiente fórmula:

$$peso(u, r) = \sum_{t=Artesano}^{Racional} Usuario(u, t) * tablaRepartoRecomendadores(t, r)$$

Donde u es el usuario y r es la estrategia de recomendación

El siguiente paso es normalizar las puntuaciones de las películas que hemos obtenido en cada una de las estrategias anteriores, para ello de cada estrategia

anterior solo se van a coger las diez mejores películas puntuadas y se van a ordenar de mayor a menor por la puntuación que hayan obtenido y su puntuación entonces quedara de la siguiente manera:

$$Puntuacion(p, r) = 10 + 1 - Posicion(p, r)$$

Donde p es la película y r es la estrategia de recomendación.

Con los datos de la puntuación y el peso el sistema ya podrá determinar cuál es la puntuación que le da a una película:

$$Puntuación(u, p) = \sum_{r=\frac{Temp}{Genero}}^{\frac{Temp}{Temp}} Puntuación(usuario, r) * peso(u, r)$$

5.2 Explicaciones para Usuarios Finales

En esta sección pondremos un ejemplo de explicaciones para el usuario final en Moodvie, para generarlas seguiremos el modelo que hemos definido en el capítulo 4 Modelo de explicación propuesto, como indicamos en ese capítulo el objetivo del modelo era generar explicaciones que aumentaran la transparencia, validez y confianza en el sistema mediante la reconstrucción de la línea de razonamiento seguida por en proceso de recomendación.

Como vimos HotSpot Explanations reconstruye la línea de razonamiento del sistema de recomendación y le genera una explicación que explique porqué debería usar ese ítem usando el razonamiento seguido por el sistema, en este caso al tratarse de un usuario final la explicación no ha de ser muy técnica, sino que lo que haremos será explicar de una manera sencilla al usuario el razonamiento usado por el recomendador.

Para el caso de Moodvie como vimos en la sección anterior, el sistema para realizar la recomendación usaba una serie de relaciones que existían entre los temperamentos del usuario, su estado de ánimo y el género de las películas. Con lo cual las explicaciones para un usuario final en Moodvie irán orientadas a explicarle porque los géneros de una película recomendada son buenos para su temperamento y estado de ánimo.

En cuanto a la generación de las explicaciones será un argumento a favor de ver la película, con lo cual solo tendremos en cuenta los aspectos positivos de los géneros con el temperamento y el estado de ánimo.

Una vez hecha una introducción a lo que queremos hacer vamos a describir todo el proceso de generación de la explicación, como vimos en la sección 4.2 el primer paso de la fase de diseño era la actividad de creación en la cual íbamos a crear un modelo que describiera el sistema de recomendación, en este caso en las Figura 5.2, 5.3, 5.4, 5.5 y 5.6 vemos el modelo de conocimiento usado por Moodvie.

Un apunte importante es que Moodvie conceptualmente es híbrido pero en realidad no existe una hibridación entre diferentes técnicas de recomendación sino que solo hay un tipo de técnica que es la basada en conocimiento, con lo cual no tendremos que combinar diferentes esquemas de modelo sino que usaremos el modelo de conocimiento genérico que definimos en la Figura 2.4 del capítulo 2, lo único que haremos será extenderlo para nuestro sistema Moodvie, el resultado como he indicado anteriormente está en las Figuras, de estas cinco figuras observamos los siguientes elementos:

Figura 5.2 Modelo de conocimiento de Moodvie 1 usuarios e ítems

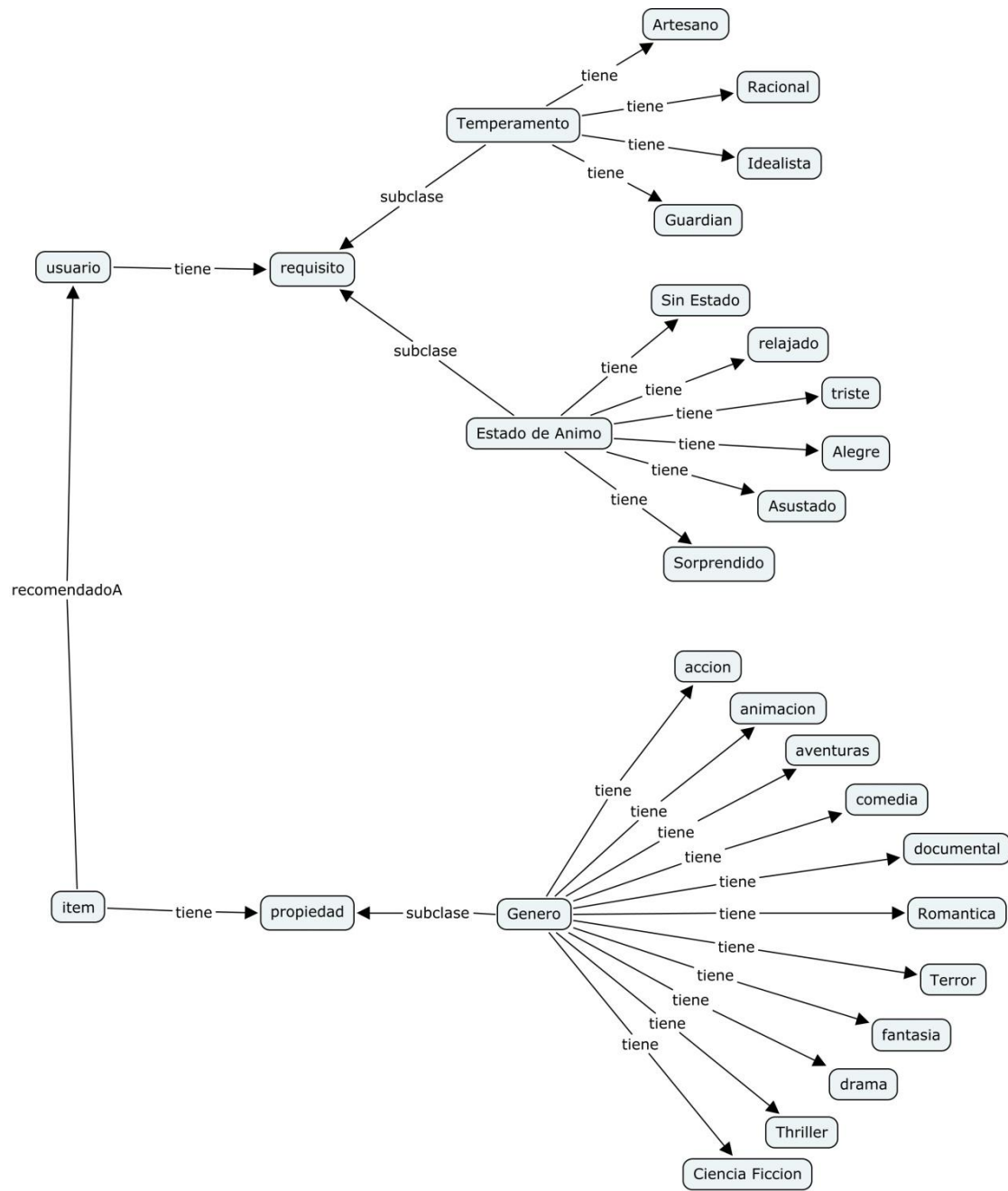


Figura 5.3 Modelo de conocimiento de Moodvie 2 relaciones entre usuarios e ítems

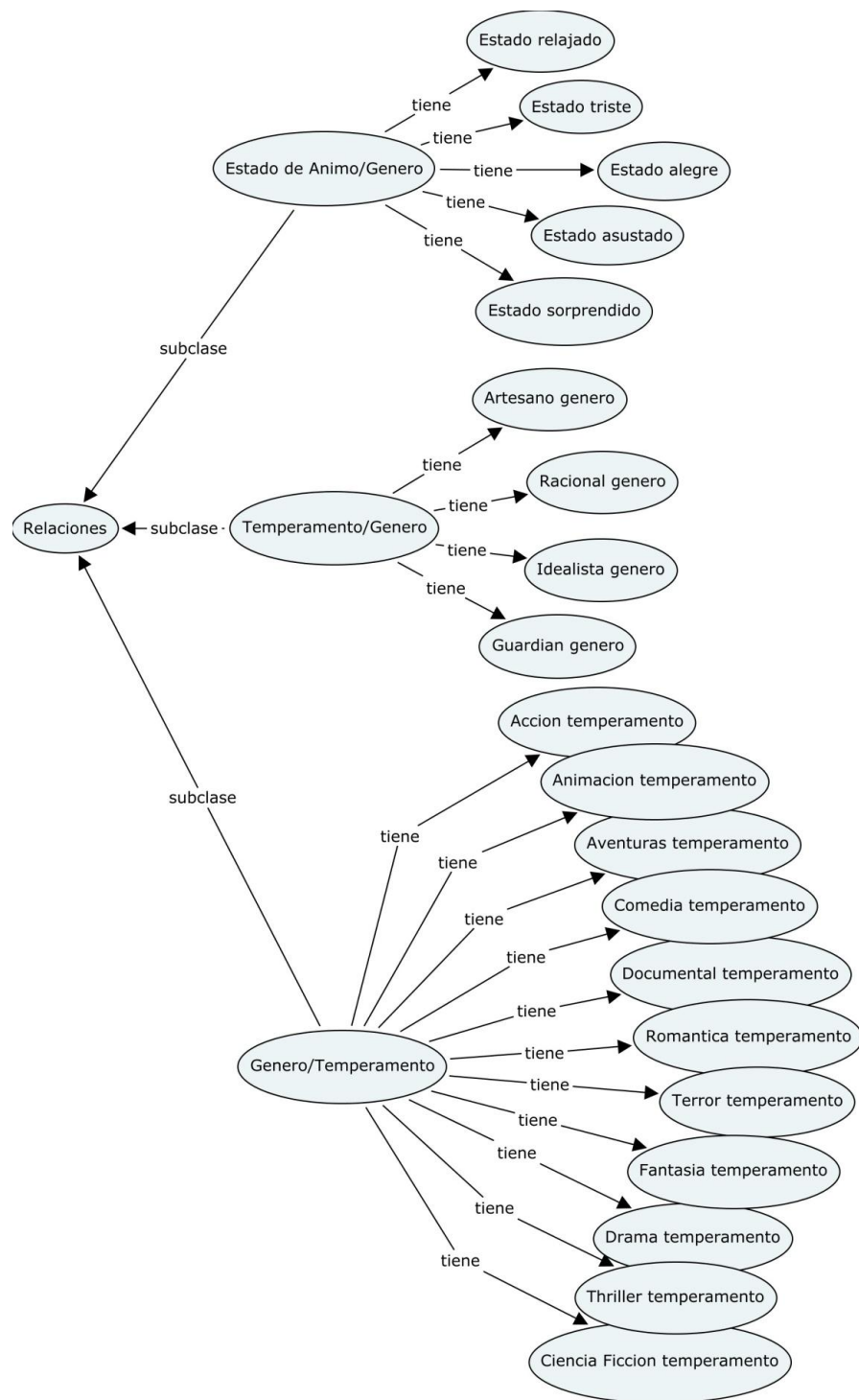


Figura 5.4 Modelo de conocimiento de Moodvie 3 relación Estado de ánimo/genero

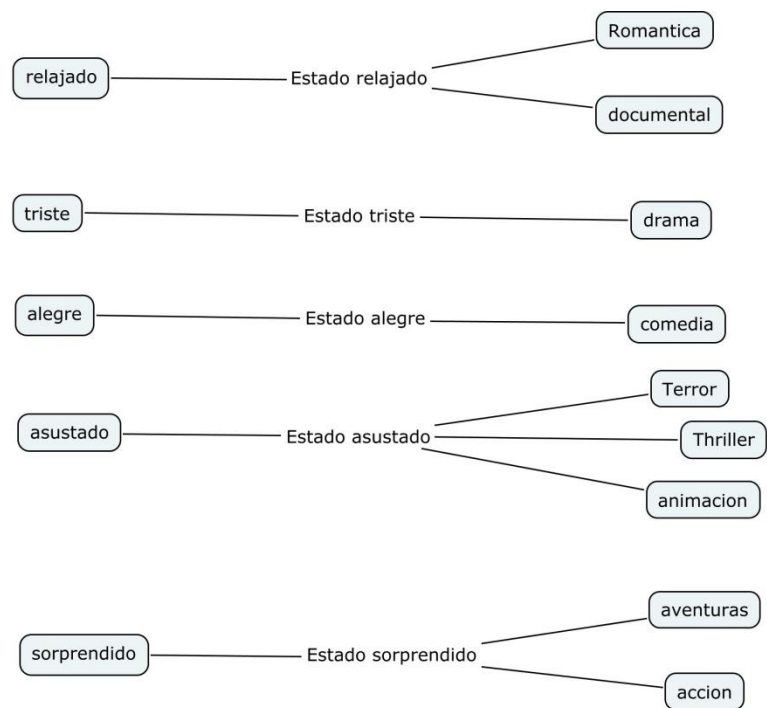


Figura 5.5 Modelo de conocimiento de Moodvie 4 relación Temperamento/Genero

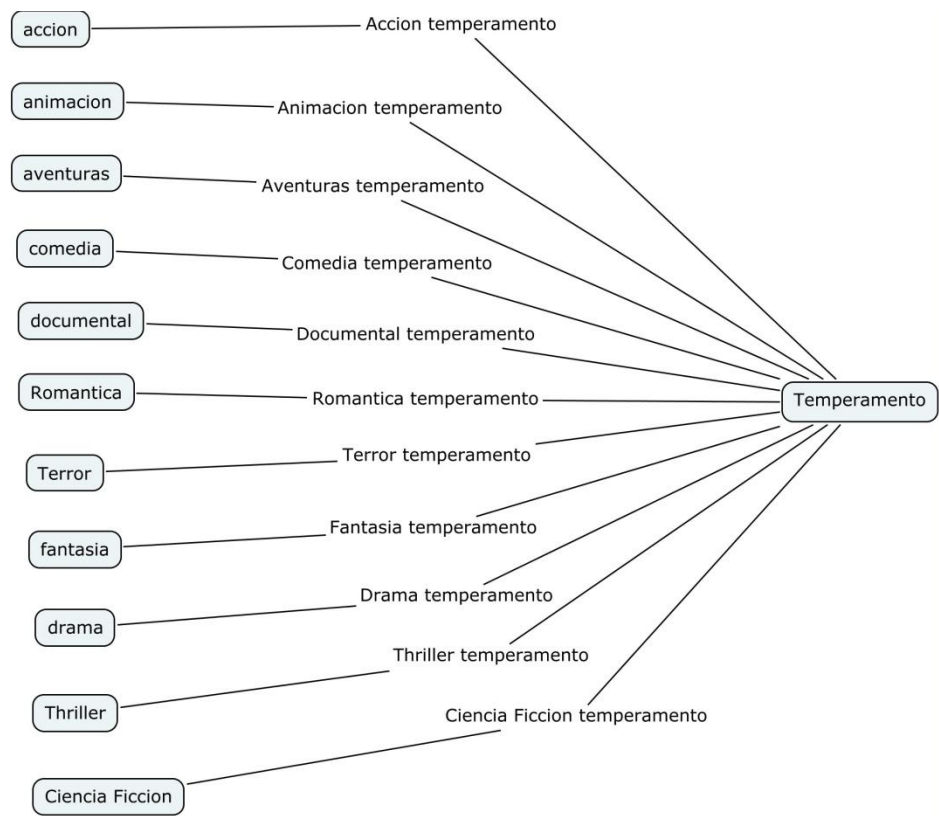
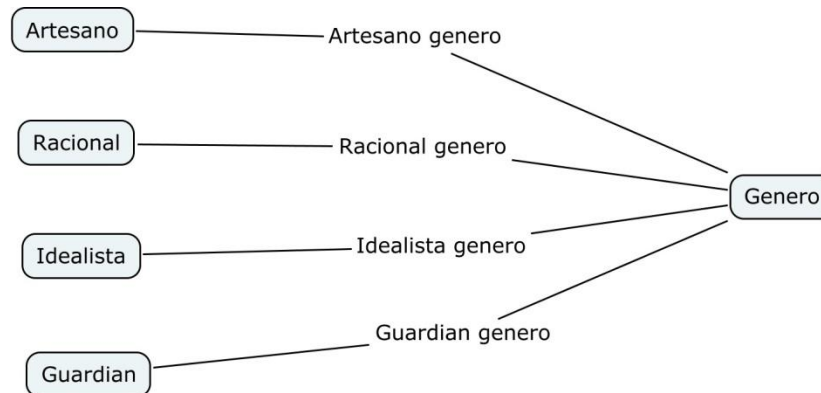


Figura 5.6 Modelo de conocimiento de Moodvie 5 relación Genero/temperamento



- Usuario, como vimos en el capítulo 2.1 en un modelo de conocimiento de una estrategia basada en conocimiento lo importante es detectar cuales son los requisitos del usuario, como se observa en la imagen tenemos dos: Los temperamentos, a su vez los temperamentos se descomponen en cuatro tipos y el otro requisito seria el estado de ánimo que pueden ser de seis tipos. Si hay alguna duda en la sección 5.1 están descritos con mayor detalle.
- Ítems, solo tendremos un tipo de ítem que será el que recomendemos al usuario, aquí lo importante será detectar cuales son las propiedades del mismo, para simplificar solo hemos reflejado las que se van a relacionar con los requisitos de usuario, en este caso los géneros de la película, este se descompondrá en los once géneros que definimos.
- Relaciones, este será el aspecto más importante de este tipo de dominios, las relaciones que existen entre los diferentes conceptos, para generar las explicaciones para el usuario final no vamos a identificar todas la relaciones sino que solo representaremos las que nos vayan a ser de utilidad para generar la explicación, para nosotros:
 - Estado de ánimo/genero, esta es la relación que definimos en la sección 5.1 en la estrategia basada en estado de ánimo, básicamente relaciona un estado de ánimo con unos determinados géneros

- Temperamento/genero, esta será la relación que definimos en la sección 5.1 en la estrategia de temperamentos, lo que hará será relacionar cada uno de los temperamentos del usuario con el género de la película.
- Genero/Temperamento, esta será la relación que definimos en la sección 5.1 en la estrategia de género, aquí lo que hará será relacionar cada uno de los géneros de la película con el temperamento del usuario.

Con todo esto, está definido el modelo del sistema de recomendación de la Figura 4.1, ahora pasaríamos a la siguiente actividad en la fase de diseño, la introspección, a partir del modelo del sistema de recomendación de la etapa anterior, de cada uno de los conceptos y relaciones que hemos determinado tendremos que identificar en qué puntos de Moodvie se encuentran para poder extraerlos mediante las trazas asociadas. En las Figuras 5.7 y 5.8 tenemos dos ejemplos de esta identificación y relación.

Con el resto de conceptos y relaciones para extraer sus datos se haría lo mismo, una vez que tenemos identificado de donde extraer el conocimiento de Moodvie y su asociación al modelo del sistema de recomendación ya tendremos definido lo que en la Figura 4.1 llamábamos el modelo de sistema de recomendación extendido.

Figura 5.7 Identificación de las trazas y su asociación con los géneros de la película en Moodvie

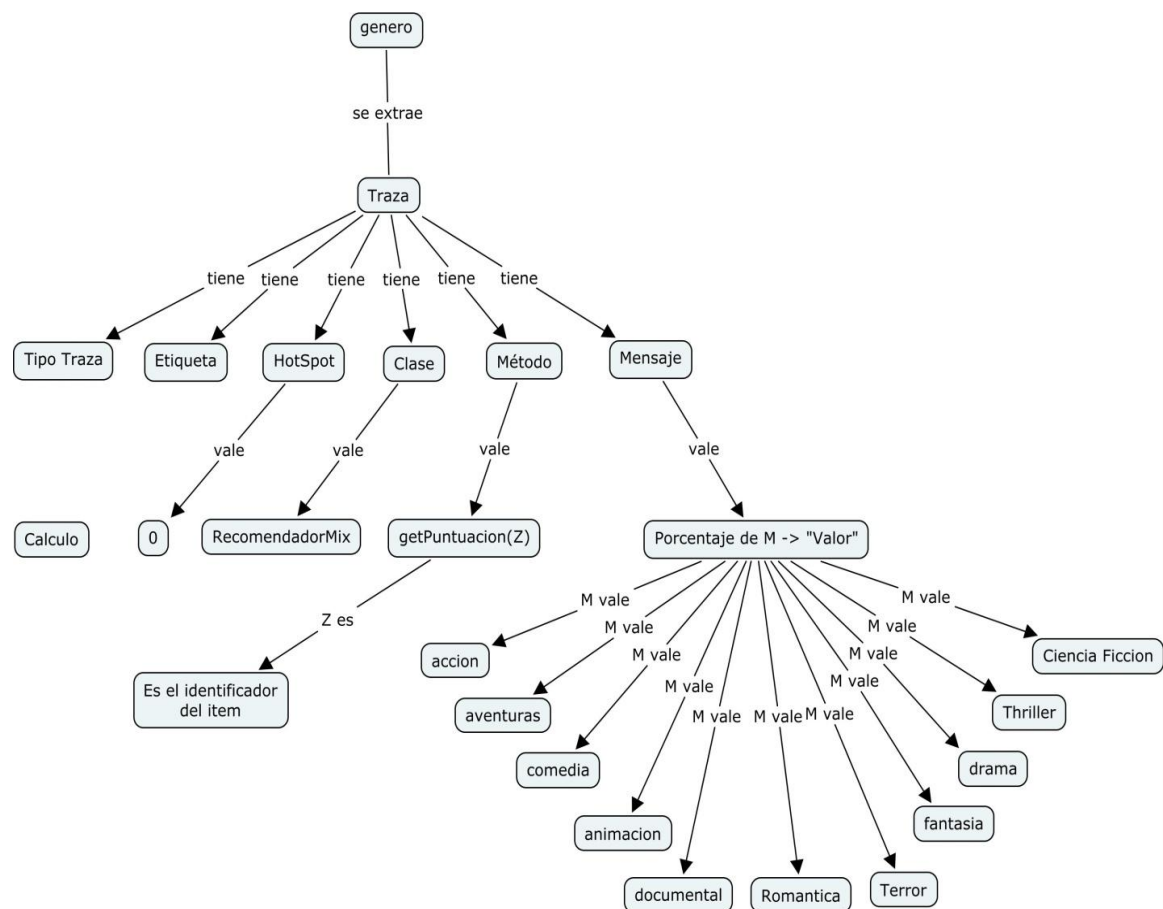
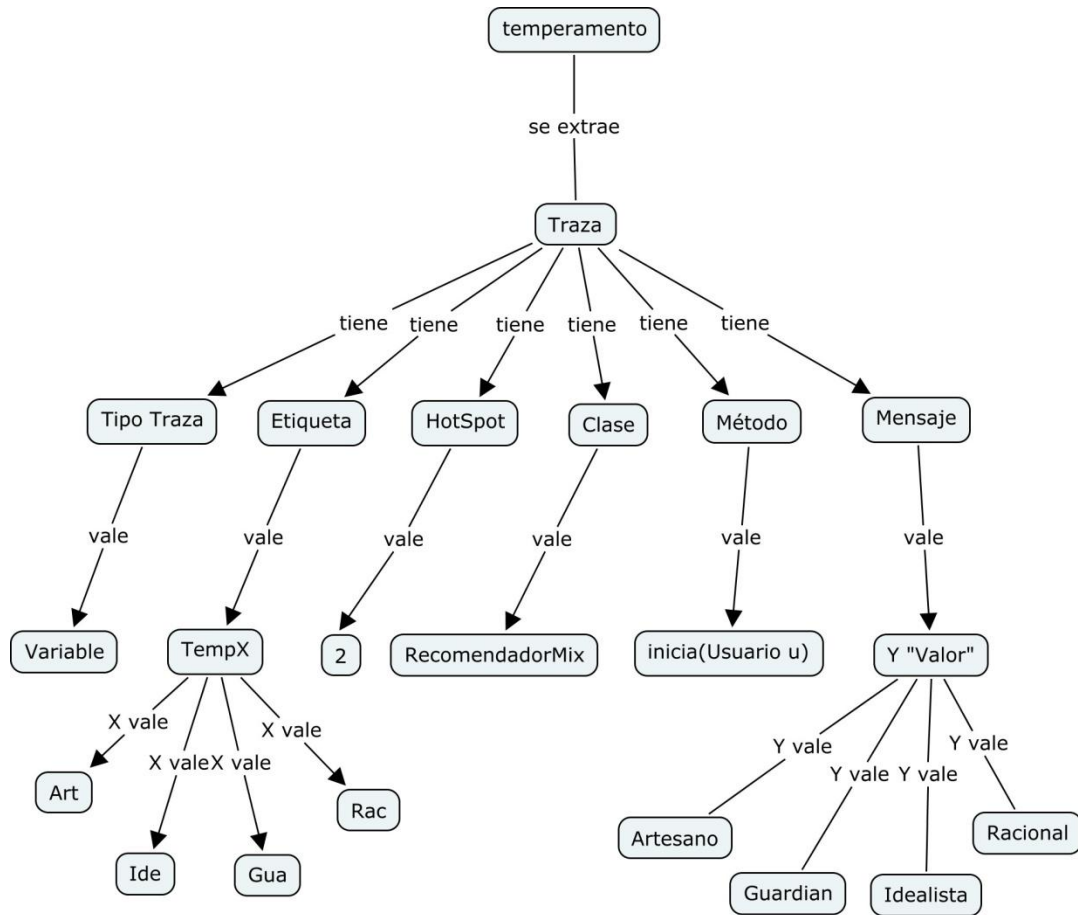
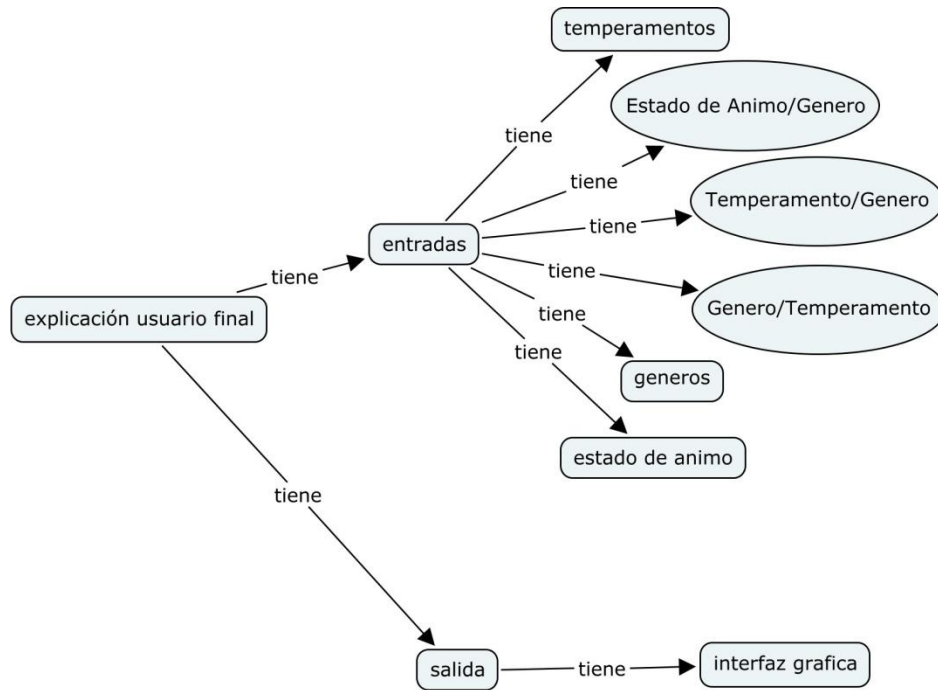


Figura 5.8 Identificación de las trazas y su asociación con los temperamentos del usuario en Moodvie



Después pasaríamos a la siguiente actividad en la fase de diseño, la explicación, a partir de uno de los modelos anteriores, es indiferente el que usemos, tendremos que crear los diferentes tipos de modelos de explicaciones que tengamos, para la creación de estos modelos usaremos la especificación del conocimiento que usamos en la sección 2.3 modelos de conocimiento de la explicación, en la Figura 5.9 tendremos el único modelo de conocimiento de la explicación que vamos a generar a los usuarios finales, donde:

Figura 5.9 Modelo de conocimiento de la explicación para usuario final en Moodvie



- Entradas, como se ve en la imagen, para poder generar la explicación necesitaremos las siguientes piezas de información del modelo del sistema de recomendación: Temperamentos y estado de ánimo del usuario, géneros de la película y todas las relaciones, en este caso al ser única la explicación va presentarle junto al usuario toda la información, pero puede existir casos en los cuales la información de entrada sí que sea más específica.
- Interfaz, representaremos la explicación de una forma grafica, en la cual se pueden ver de una forma conjunta las explicaciones referentes al estado de ánimo con el género y temperamento con genero.

El proceso, esto lo describiremos de una forma textual es lo único que no queda reflejado en el modelo de conocimiento, la interfaz quedara dividida en dos zonas:

1. En la primera parte se explicara lo referente al estado de ánimo, aquí pueden suceder dos cosas:
 - El usuario no selecciona estado de ánimo, con lo cual no se genera explicación de este tipo, la explicación se van centrar únicamente en el temperamento del usuario

- El usuario ha seleccionado estado de ánimo. En este paso la explicación a su vez dependerá del estado anímico que quiera obtener el usuario, pudiendo ser:
 - Sorprendido. En este caso la explicación que se mostrará en su zona asociada de la interfaz grafica será de la siguiente forma:
“La película recomendada es ideal para sorprenderte porque tiene unos porcentajes altos en los géneros acción y aventura”
 - Alegre. En este caso la explicación que se mostrará en su zona asociada de la interfaz grafica será de la siguiente forma:
“La película recomendada es ideal para hacerte reír porque tiene unos porcentajes altos en el género de comedia”
 - Asustado. En este caso la explicación que se mostrará en su zona asociada de la interfaz grafica será de la siguiente forma:
“La película recomendada es ideal para hacerte asustarte porque tiene unos porcentajes altos en los géneros terror y thriller”
 - Triste. En este caso la explicación que se mostrará en su zona asociada de la interfaz grafica será de la siguiente forma:
“La película recomendada es ideal para entristecerte porque tiene unos porcentajes altos en el género drama”
 - Relajado. En este caso la explicación que se mostrará en su zona asociada de la interfaz grafica será de la siguiente forma:
“La película recomendada es ideal para relajarte porque tiene unos porcentajes altos en el género drama”

2. En la segunda parte se explicara todo lo relacionado con el temperamento del usuario. Esta es la explicación más complicada de generar, la relación existente entre el temperamento del usuario y los géneros de las películas, se centrara en los aspectos positivos de estas relaciones de forma que la explicación sea un argumento positivo a favor de ver la película. Como vimos en la relaciones tendremos los datos asociados a las relaciones:

- a. Temperamento/Genero. El dato que extraemos de esta relación es un valor que me dice la similitud existente entre un determinado género de un temperamento y el género asociado a esa película.
- b. Genero/Temperamento. El dato que extraemos de esta relación es un valor que me dice la similitud existente entre un determinado temperamento de una película y el temperamento asociado del usuario.

Una vez que conocemos estas dos similitudes, tenemos que seleccionar cuales son los aspectos más positivos para la explicación, para ello determinaremos cual es la calidad de cada uno de ellos, mediante las siguientes formulas:

$$calidad(genero) = similitudTemperamentoGenero(genero)$$

$$calidad(temperamento) = similitudGeneroTemperamento(Temperamento)$$

Una vez tenemos la calidad tanto de los temperamentos como de los géneros, ahora toca seleccionar cuales mostraremos al usuario, para simplificar le mostraremos al usuario los cuatro mejores

De forma que la explicación que se le muestre al usuario tendrá la siguiente forma, dependiendo de si es un temperamento o un género y del valor de la calidad, el pseudocódigo es el siguiente:

```

Si calidad(x) > 95 Entonces
    Si x = temperamento Entonces
        "Los géneros de la película se adaptan perfectamente a tu
        porcentaje de temperamentoX"
    Si no Entonces
        "Tu temperamento se adaptan perfectamente al porcentaje de
        generoX"
Si no si calidad(x) > 90 Entonces
    Si x = temperamento Entonces
        "Los géneros de la película se adaptan muy bien a tu porcentaje de
        temperamentoX"
    Si no Entonces
        "Tu temperamento se adaptan muy bien al porcentaje de generoX"
Si no si calidad(x) > 80 Entonces
    Si x = temperamento Entonces
        "Los géneros de la película se adaptan bien a tu porcentaje de
        temperamentoX"
    Si no Entonces

```

```

        "Tu temperamento se adaptan bien al porcentaje de generoX"
Si no
    Si  $x = \text{temperamento}$  Entonces
        "Los géneros de la película se adaptan más o menos bien a tu
        porcentaje de temperamentoX"
    Si no Entonces
        "Tu temperamento se adaptan más o menos bien al porcentaje de
        generoX"

```

5.3 Explicaciones para Ingeniero del Conocimiento

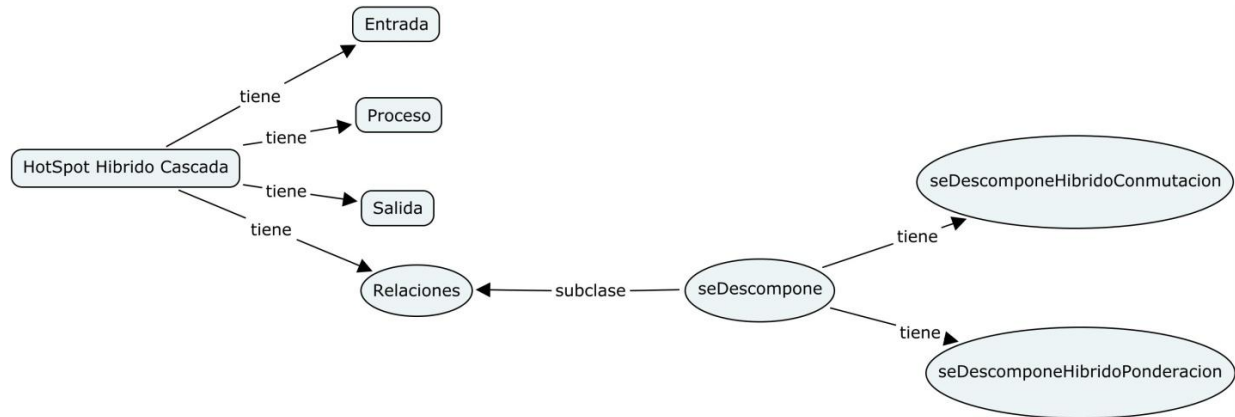
En esta sección pondremos un ejemplo de explicaciones para el ingeniero del conocimiento en Moodvie, para generarlas seguiremos el modelo que definimos en el capítulo 4 Modelo de explicación propuesto, como indicamos en ese capítulo el objetivo del modelo para este tipo de usuarios era el de validar el sistema mediante la reconstrucción de la línea de razonamiento seguida en el proceso de recomendación.

Como vimos HotSpotExplanations reconstruye la línea de razonamiento del sistema de recomendación y en este caso le permite generar una explicación al ingeniero del conocimiento de cómo se ha realizado el proceso de recomendación para ese ítem, en este caso al tratarse del ingeniero del conocimiento la explicación le mostrara los datos del proceso de recomendación en los cuales el ingeniero este interesado en verificar.

Como vimos en el capítulo 4, en el modelo de la fase de diseño la primera actividad es la creación, en la cual generaremos un modelo del sistema de recomendación, en este caso para generar este modelo utilizaremos el modelo de conocimiento que definimos en la sección 2.2 Modelo de conocimiento para ingeniero del conocimiento, lo fundamental es identificar los puntos calientes que se quieran verificar, una vez hayamos detectado los puntos calientes tenemos que identificar los datos queremos verificar y clasificarlos atendiendo a si son: entradas del punto caliente, el proceso del punto caliente, las salidas del mismo y como se relaciona con los otros puntos calientes. Los puntos calientes que estamos interesados en verificar y generar explicaciones de Moodvie son los siguientes:

1. Híbrida en cascada, ver Figura 5.10.

Figura 5.10 Modelo de Conocimiento del punto caliente híbrido en cascada

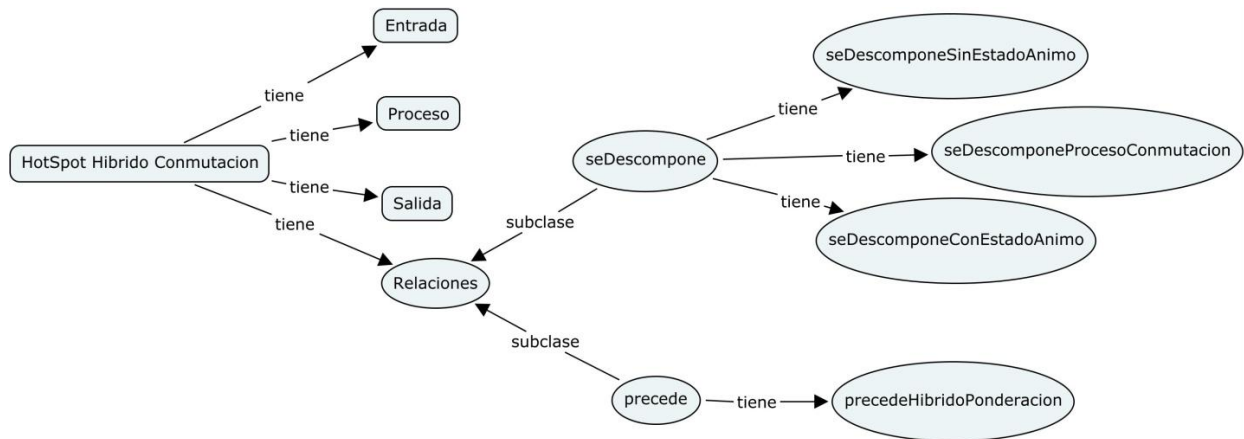


De la figura observamos los siguientes elementos:

- Relaciones, tendremos dos relaciones y serán la de descomposición del proceso de recomendación de la hibridación en cascada, como vimos se descomponía en hibridación en conmutación e hibridación en ponderación.

2. Hibrida en conmutación, ver Figura 5.11.

Figura 5.11 Modelo de Conocimiento del punto caliente híbrido en conmutación



De la figura observamos los siguientes elementos:

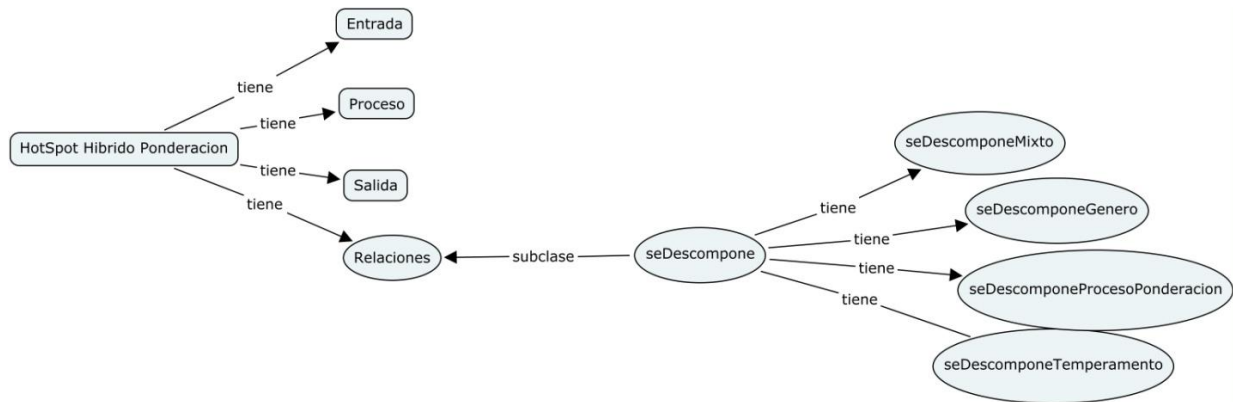
- Relaciones, tendremos cuatro relaciones y serán:
 - Tres correspondientes a la descomposición del proceso de funcionamiento del recomendador híbrido en conmutación, dos de

ellas correspondientes a las estrategias que combina y una correspondiente al proceso de conmutación que determina entre cual ha de elegir.

- Una correspondiente a la relación de precedencia de esta hibridación respecto a la de ponderación.

3. Hibrida en ponderación, ver Figura 5.12.

Figura 5.12 Modelo de Conocimiento del punto caliente híbrido en ponderación

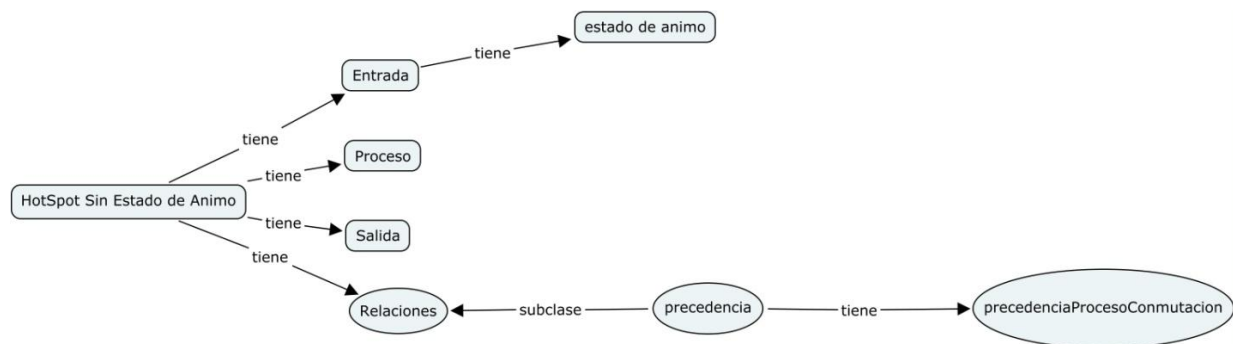


De la figura observamos los siguientes elementos:

- Relaciones, tendremos cuatro relaciones y serán la de descomposición del proceso de recomendación la hibridación ponderada, tres de ellas referentes a las estrategias que quiere hibridar y una referente al proceso de ponderación que combina las tres estrategias anteriores.

4. Estrategia sin estado de ánimo, ver Figura 5.13

Figura 5.13 Modelo de Conocimiento del punto caliente sin estado de ánimo

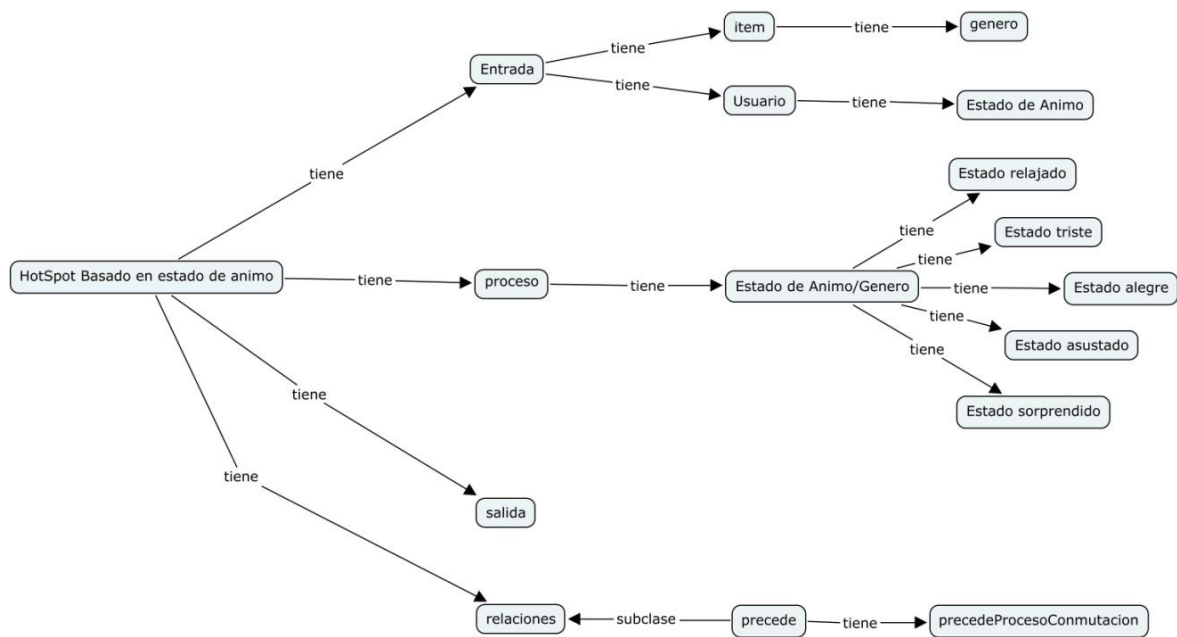


De la figura observamos los siguientes elementos:

- Entrada, en este punto caliente los datos de entrada que queremos verificar es el estado de ánimo del usuario como los géneros de las películas.
- Relaciones, tendremos una relación de precedencia y es que esta estrategia precede al proceso de conmutación.

5. Estrategia con estado de ánimo, ver Figura 5.14.

Figura 5.14 Modelo de Conocimiento del punto caliente estrategia basada en estado de ánimo



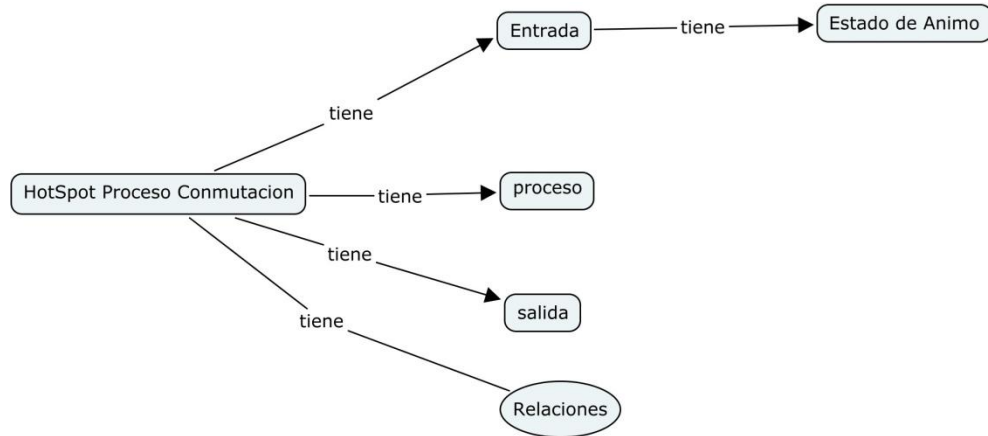
De la figura observamos los siguientes elementos:

- Entrada, en este punto caliente los datos de entrada que queremos verificar es el estado de ánimo del usuario como los géneros de las películas.
- Proceso, aquí lo que nos interesa verificar es en el caso que haya habido recomendación basada en estado de ánimo que se hayan aplicado correctamente las relaciones que hay entre los estado de ánimo y el género, estas relaciones están descrita en la sección 5.1 estrategia basada en estado de ánimo.

- Relaciones, tendremos una relación de precedencia y es que esta estrategia precede al proceso de conmutación.

6. Proceso conmutación, ver Figura 5.15.

Figura 5.15 Modelo de Conocimiento del punto caliente proceso de conmutación

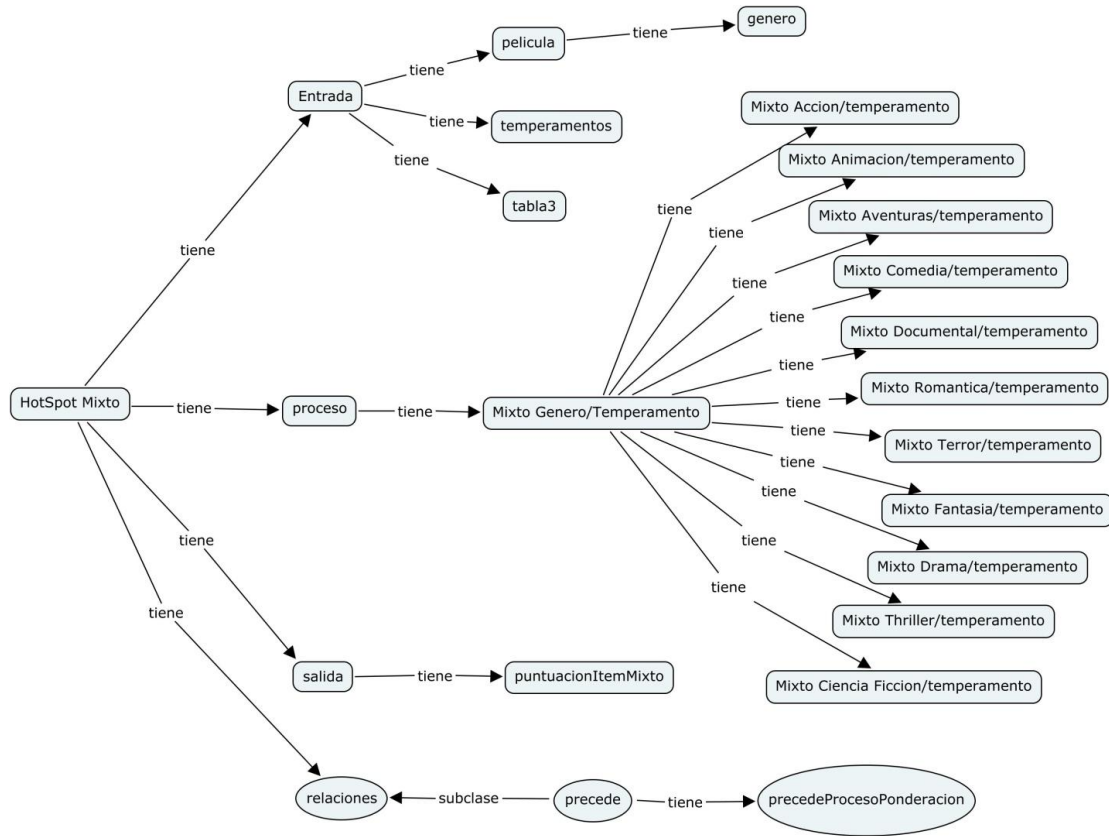


De la figura observamos los siguientes elementos:

- Entradas, en este punto caliente los datos de entrada que queremos verificar es el estado de ánimo, este es muy importante ya que nos permitirá decidir cuál de las dos estrategias seleccionar.

7. Estrategia mixta, ver Figura 5.16.

Figura 5.16 Modelo de Conocimiento del punto caliente estrategia mixta

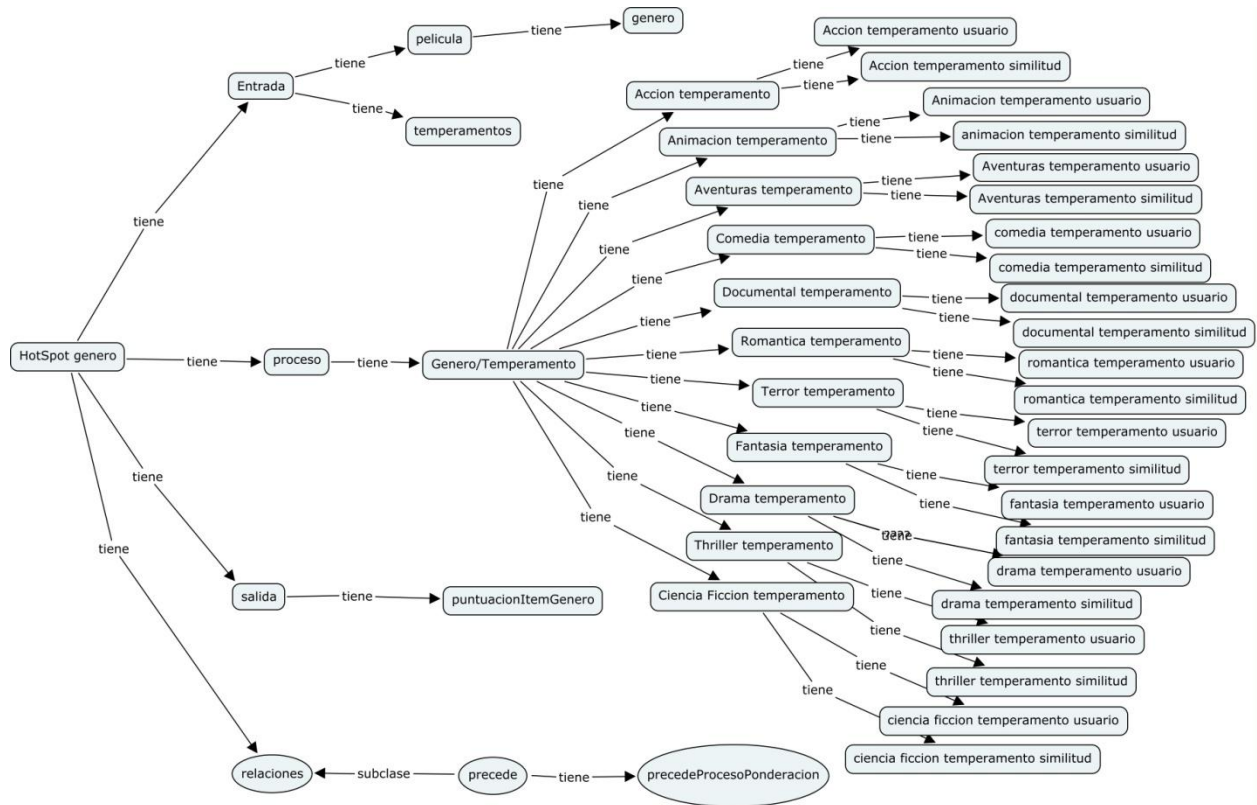


De la figura observamos los siguientes elementos:

- Entradas, en este punto caliente los datos de entrada que queremos verificar son los temperamentos del usuario, géneros de las películas y la Tabla 5.4 que vimos en la sección 5.1.
- Proceso, en el proceso nos interesa verificar los datos referentes a las relaciones que hay entre los temperamentos del usuario y el género de una película, como vimos en la sección 5.1 estrategia mixta. Para cada género de la película almacenaremos el valor de la combinación de la *tabla3* con los temperamentos del usuario, el valor del porcentaje de género de una película ya lo conocemos de la entrada.
- Salida, nos quedaremos con el valor de la puntuación final que da la estrategia a la película que recibe por la entrada para su verificación.
- Relaciones, tendremos tan solo una relación y será la de precedencia con el proceso de ponderación.

8. Estrategia genero, ver Figura 5.17.

Figura 5.17 Modelo de Conocimiento del punto caliente estrategia basada en genero



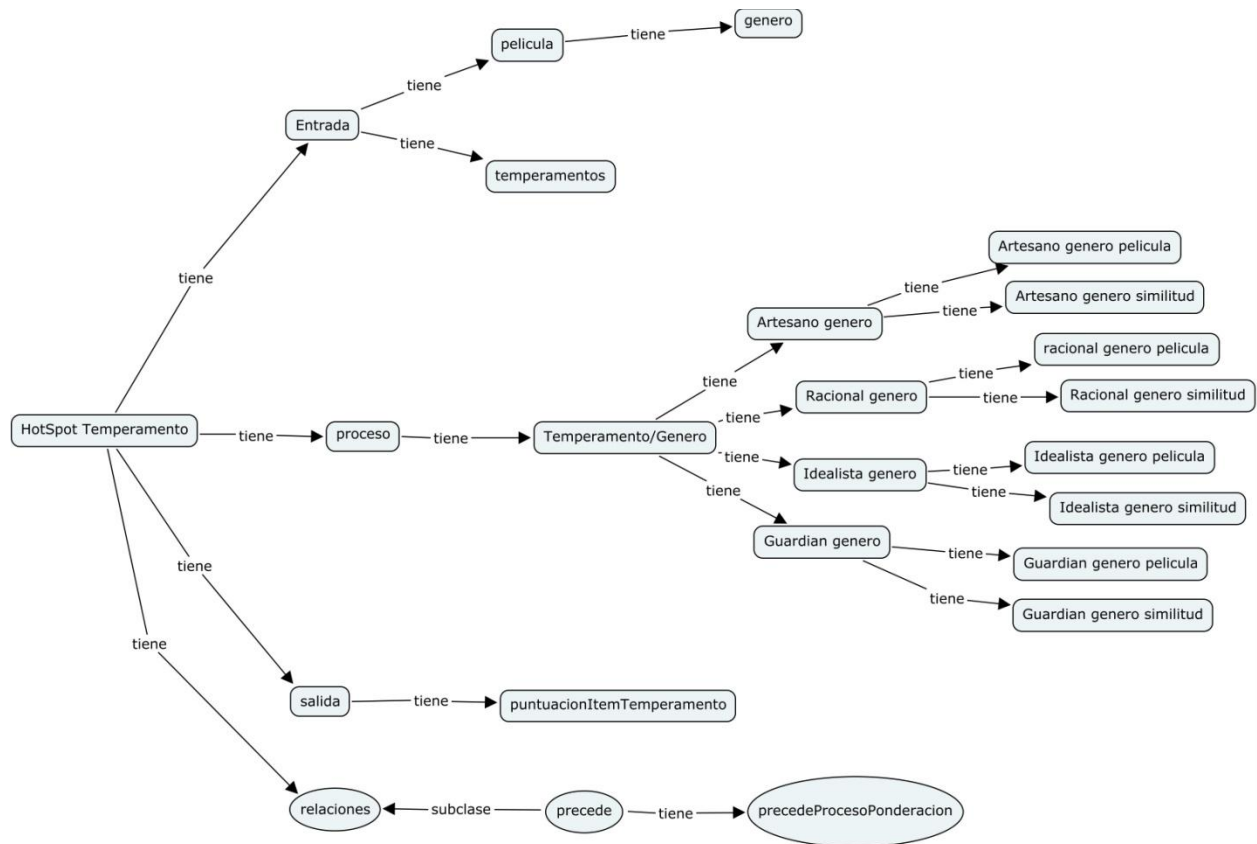
De la figura observamos los siguientes elementos:

- Entradas, en este punto caliente los datos de entrada que queremos verificar son tanto los temperamentos del usuario como los géneros de las películas.
- Proceso, en el proceso nos interesa verificar los datos referentes a las relaciones que hay entre los géneros de un usuario y los géneros de la película, como vimos en la sección 5.1 estrategia basada en el género. Para cada género del usuario almacenaremos su valor y su similitud con el género de la película, los géneros de la película ya lo recibimos por la entrada.

- Salida, nos quedaremos con el valor de la puntuación final que da la estrategia a la película que recibe por la entrada para su verificación.
- Relaciones, tendremos tan solo una relación y será la de precedencia con el proceso de ponderación.

9. Estrategia temperamento, ver Figura 5.18.

Figura 5.18 Modelo de Conocimiento del punto caliente estrategia temperamento



De la figura observamos los siguientes elementos:

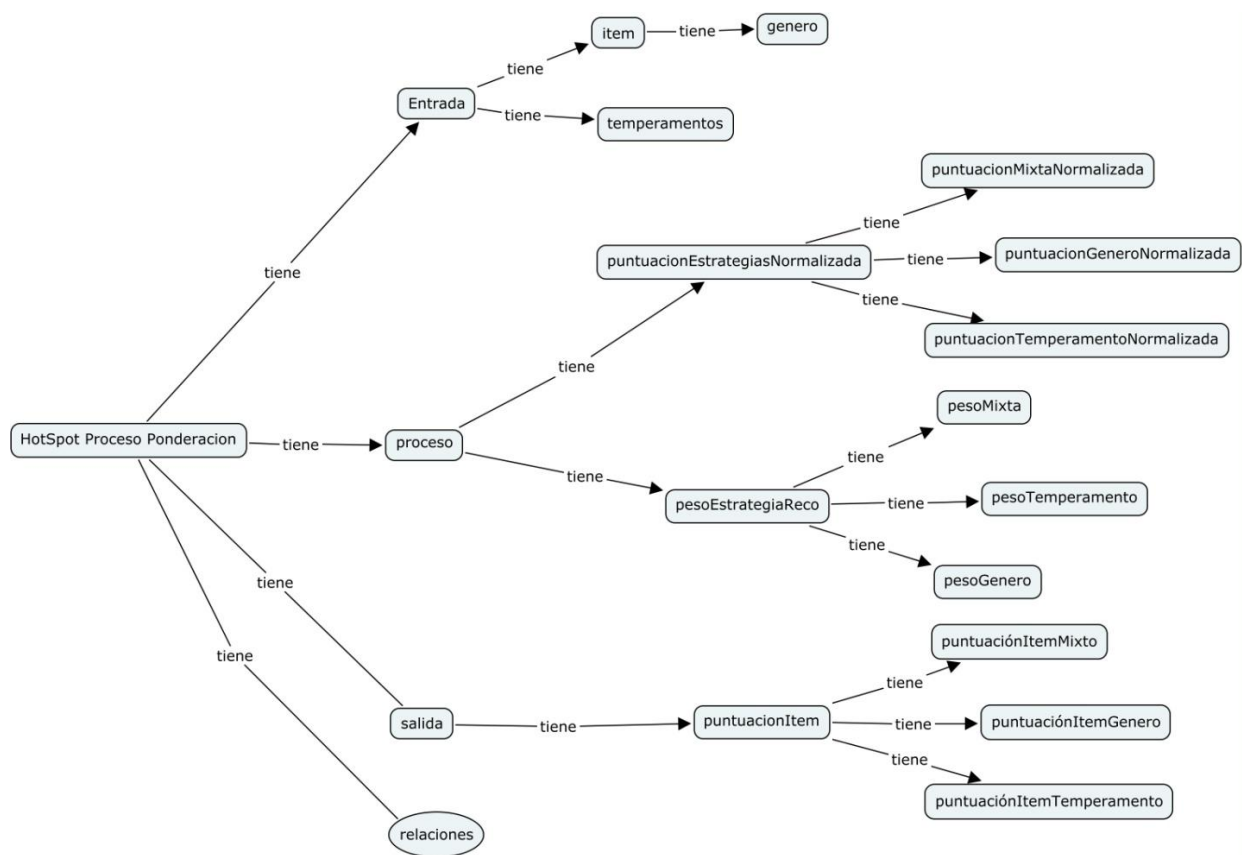
- Entradas, en este punto caliente los datos de entrada que queremos verificar son tanto los temperamentos del usuario como los géneros de la películas
- Proceso, en el proceso nos interesa verificar los datos referentes a las relaciones que hay entre los temperamentos de una película y el temperamento de un usuario, como vimos en la sección 5.1 estrategia

basada en temperamento. Para cada temperamento de la película almacenaremos su valor y su similitud con el temperamento del usuario, el temperamento del usuario ya lo recibimos por la entrada.

- Salida, nos quedaremos con el valor de la puntuación final que da la estrategia a la película que recibe por la entrada para su verificación.
- Relaciones, tendremos tan solo una relación y será la de precedencia con el proceso de ponderación.

10. Proceso ponderación, ver Figura 5.19.

Figura 5.19 Modelo de Conocimiento del punto caliente proceso ponderación



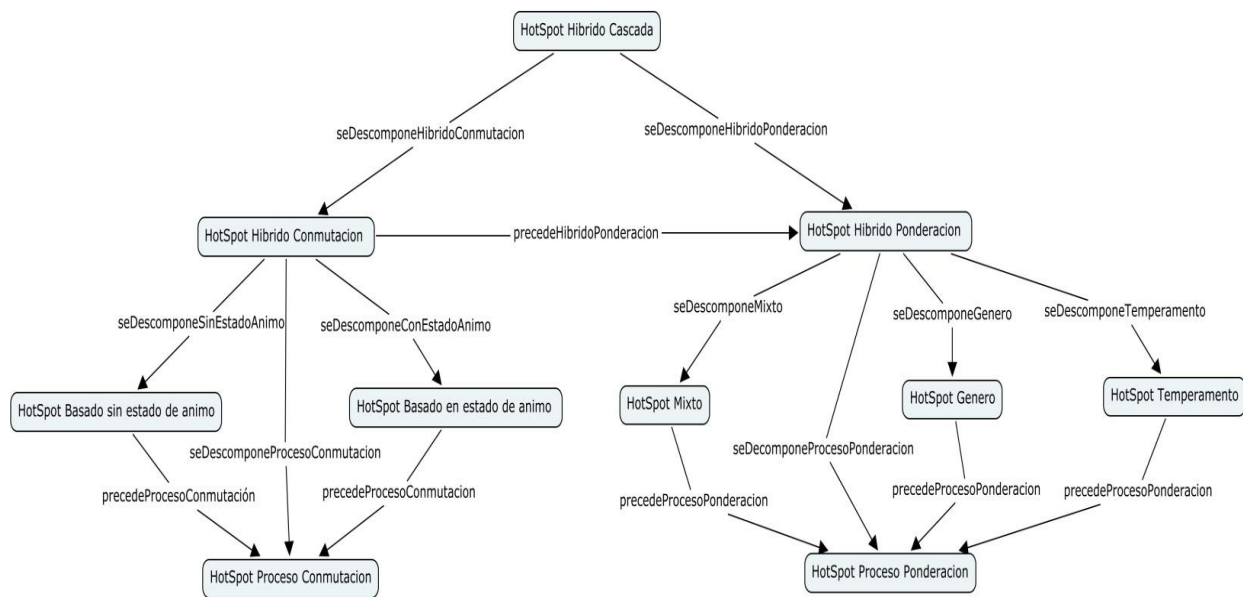
De la figura observamos los siguientes elementos:

- Entradas, en este punto caliente los datos de entrada que queremos verificar son tanto los temperamentos del usuario como los géneros de las películas.

- Proceso, en el proceso nos interesa verificar los datos al proceso de ponderación como vimos en la sección 5.1. Aquí lo que nos interesa verificar es tanto la puntuación normalizada que ha obtenido el ítem en cada una de las diferentes estrategias, esto es un valor de uno a diez, y por otro lado cual ha sido el peso que tiene cada estrategia de recomendación en función de los temperamentos del usuario.
- Salida, nos quedaremos con el valor de la puntuación final que da la estrategia a la película que recibe por la entrada para su verificación.

En la siguiente Figura 5.20 Mostraremos una visión general de los puntos calientes, en la cual se puede observar cómo se relacionan todos los puntos calientes entre sí.

Figura 5.20 Relaciones entre los puntos calientes identificados en Moodvie



Una vez que hemos determinado como son los modelo de conocimiento de los puntos calientes que queremos verificar, con todo esto ya estaría construido nuestro modelo del sistema de recomendación, ahora pasaríamos a la siguiente actividad en la fase de diseño, la introspección, a partir del modelo del sistema de recomendación de la etapa anterior, de cada uno de los conceptos que hemos determinado tendremos que identificar en qué puntos de Moodvie se encuentran para poder extraerlos mediante las trazas asociadas. En las Figuras 5.7 y 5.8 tenemos dos ejemplos de esta identificación y relación.

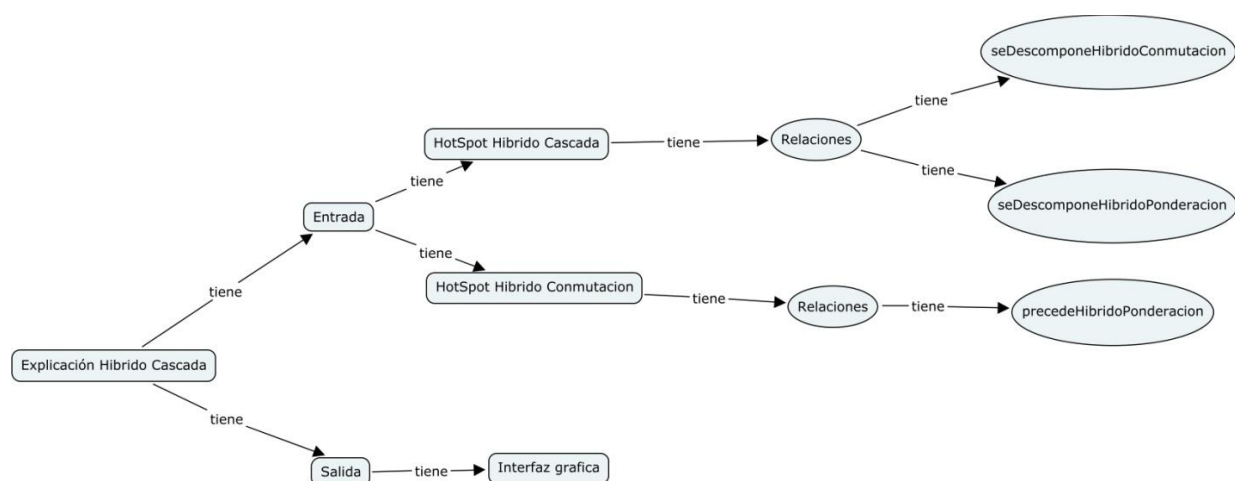
Con el resto de conceptos y relaciones para extraer sus datos se haría lo mismo, una vez que tenemos identificado de donde extraer el conocimiento de Moodvie y su asociación al modelo del sistema de recomendación ya tendremos definido lo que en la Figura 4.1 llamábamos el modelo de sistema de recomendación extendido.

Como hemos visto en los anteriores modelo de conocimiento hay muchos datos de entrada que se son siempre los mismos, sobre todo los referentes a los temperamentos y géneros, lo ideal es que si se quiere obtener una verificación total, las trazas se tendrían que extraer del punto del programa al que este asociado ese punto caliente.

La siguiente actividad en la fase de diseño es la explicación, a partir de uno de los modelo anteriores, es indiferente el que usemos, tendremos que crear los diferentes tipos de modelos de explicaciones que tengamos, para la creación de estos modelos usaremos la especificación del conocimiento que usamos en la sección 2.3 Modelo de conocimiento de la explicación, nosotros hemos optado en este caso que por cada punto caliente que hemos identificado generaremos una explicación asociada, con lo cual tendremos:

El primer modelo de conocimiento de la explicación será el que esté relacionado con el punto caliente de la hibridación en cascada, en la Figura 5.21 tenemos el modelo de conocimiento de esta explicación, donde:

Figura 5.21 Modelo de conocimiento de la explicación para el punto caliente híbrido en cascada

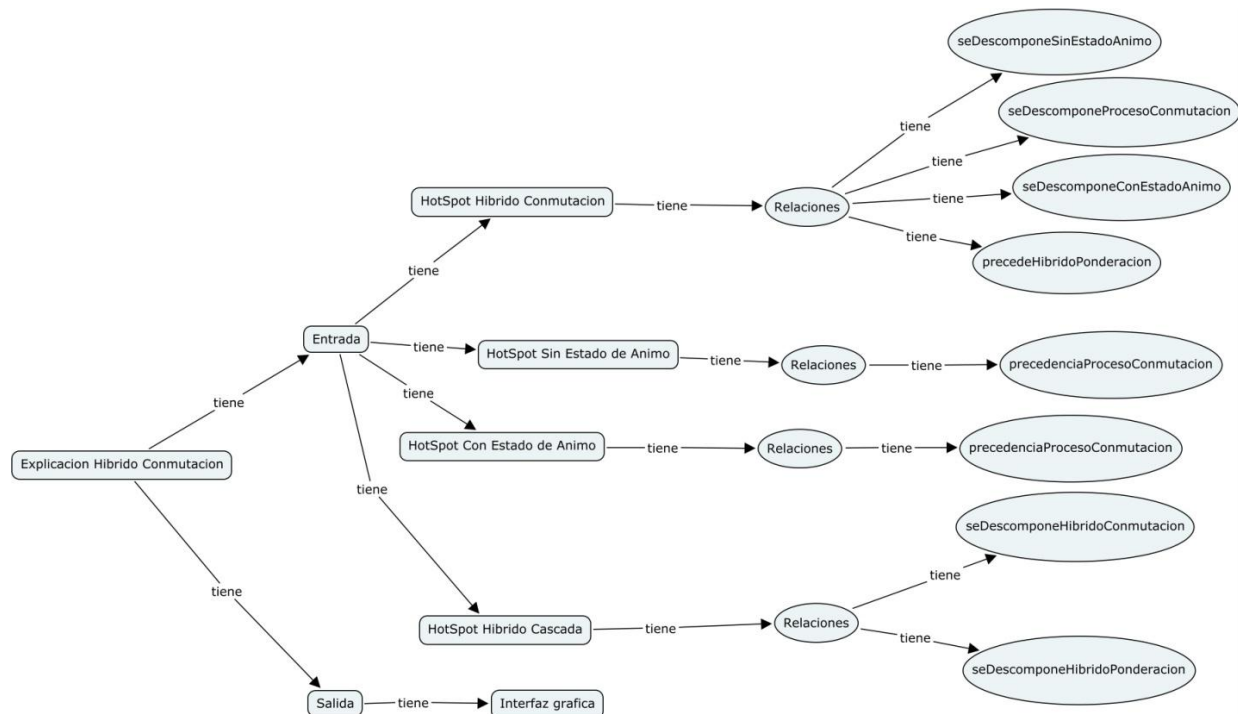


- Entradas, como se ve en la imagen, para poder generar la explicación necesitaremos las siguientes piezas de información del modelo del sistema de recomendación: Las relaciones de descomposición del proceso de funcionamiento del recomendador híbrido en cascada y de precedencia del punto caliente de hibridación en conmutación para saber cuál de las dos se ejecuta antes.
- Interfaz, representaremos la explicación de una forma grafica, en la cual se puede observar en las dos estrategias de recomendación en las que se divide la hibridación en cascada.

El proceso, en esta explicación no habrá cálculos intermedios, estas explicaciones tan solo serán interfaces graficas que servirán de apoyo al ingeniero del conocimiento para situarse mejor en las diferentes estrategias y además le permitirá la navegación entre las mismas, de esta forma se le facilitara la depuración del sistema de recomendación.

El segundo modelo de conocimiento de la explicación será el que esté relacionado con el punto caliente de la hibridación en conmutación, en la Figura 5.22 tenemos el modelo de conocimiento de esta explicación, donde:

Figura 5.22 Modelo de conocimiento de la explicación para el punto caliente híbrido en conmutación

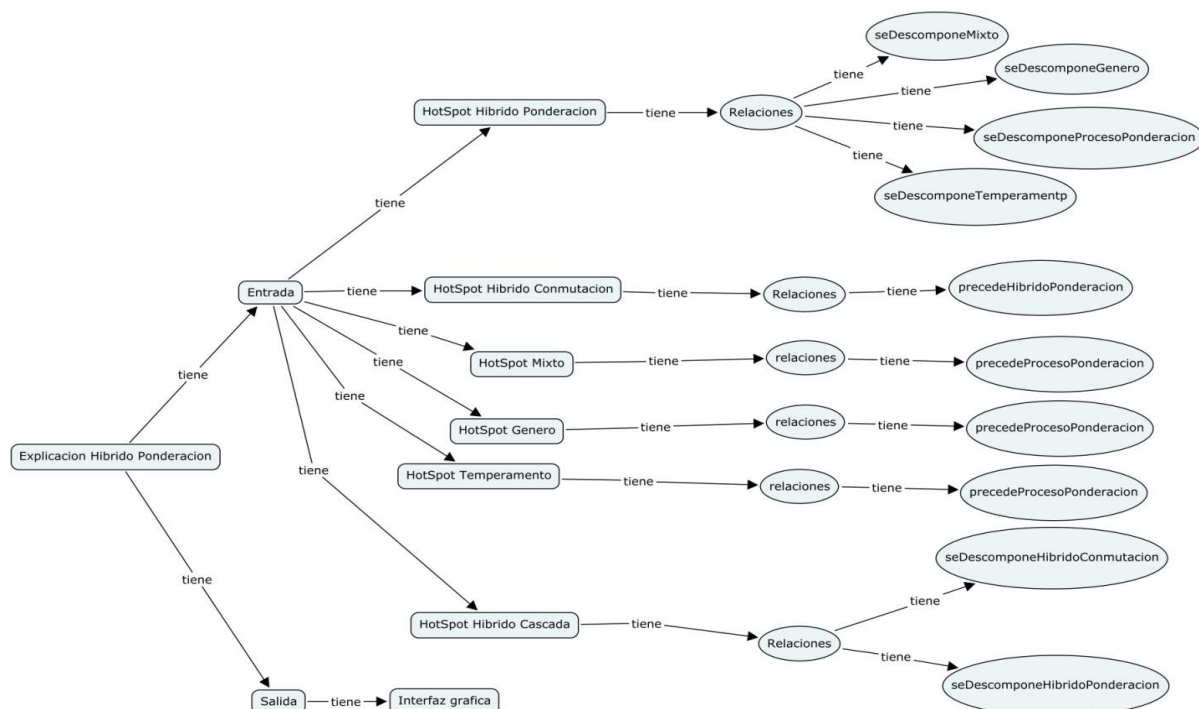


- Entradas, como se ve en la imagen, para poder generar la explicación necesitaremos las siguientes piezas de información del modelo del sistema de recomendación: Las relaciones de descomposición del proceso de funcionamiento del recomendador híbrido en cascada, las de descomposición del proceso de funcionamiento del recomendador híbrido en conmutación, las de precedencia respecto a la de hibridación en ponderación y los precedencia entre los diferentes procesos en los que ha sido descompuesto la hibridación en conmutación.
- Interfaz, representaremos la explicación de una forma grafica, en la cual se puede observar en las dos estrategias de recomendación en las que se divide la hibridación en ponderación y el proceso de conmutación.

El proceso, seguirán la misma idea que las de hibridación en cascada.

El tercer modelo de conocimiento de la explicación será el que esté relacionado con el punto caliente de la hibridación en ponderación, en la Figura 5.23 tenemos el modelo de conocimiento de esta explicación, donde:

Figura 5.23 Modelo de conocimiento de la explicación para el punto caliente híbrido en ponderación

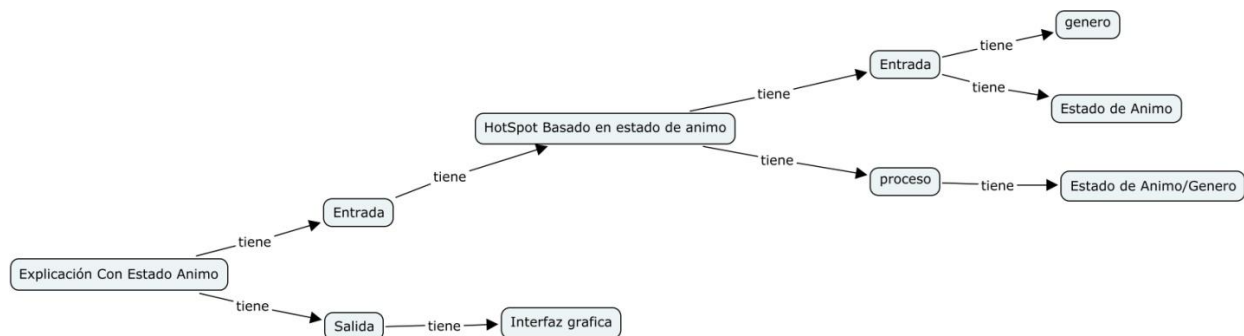


- Entradas, como se ve en la imagen, para poder generar la explicación necesitaremos las siguientes piezas de información del modelo del sistema de recomendación: Las relaciones de descomposición del proceso de funcionamiento del recomendador híbrido en cascada, las de descomposición del proceso de funcionamiento del recomendador híbrido en ponderación, las de descomposición del proceso de funcionamiento del recomendador híbrido en ponderación, las de precedencia respecto a la de hibridación en ponderación y las de precedencia entre los diferentes procesos en los que ha sido descompuesto la hibridación en ponderación.
- Interfaz, representaremos la explicación de una forma grafica, en la cual se puede observar en las tres estrategias de recomendación en las que se divide la hibridación en ponderación y el proceso de ponderación asociado.

El proceso, seguirán la misma idea que las de hibridación en cascada.

El cuarto modelo de conocimiento de la explicación será el que esté relacionado con el punto caliente del estado de ánimo, esta explicación solo estará disponible si el ingeniero ha usado la recomendación basada en estado de ánimo al hacer la recomendación, en la Figura 5.24 tenemos el modelo de conocimiento de esta explicación, donde:

Figura 5.24 Modelo de conocimiento de la explicación para el estado de ánimo



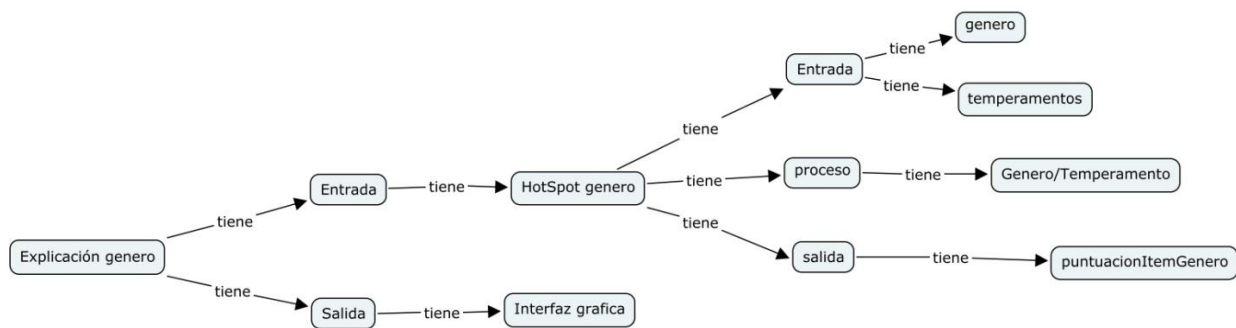
- Entradas, como se ve en la imagen, para poder generar la explicación necesitaremos las siguientes piezas de información del modelo del sistema de recomendación: del punto caliente basado en estado de ánimo necesitaremos tantos sus entradas como su proceso.

- Interfaz, representaremos la explicación de una forma grafica, en la cual existirá una explicación que permita verificar que los géneros de la película se adaptan correcta al estado de ánimo del usuario.

El proceso, en este caso no habrá ningún tipo de cálculo intermedio, tan solo se le presentara al ingeniero dependiendo del estado de ánimo los géneros que están afectados en la película y si cumplen la relación entre ellos.

El quinto modelo de conocimiento de la explicación que veremos será el que esté relacionado con el punto caliente del género, en la Figura 5.25 Tenemos el modelo de conocimiento de esta explicación, donde:

Figura 5.25 Modelo de conocimiento de la explicación para la estrategia basada en el género

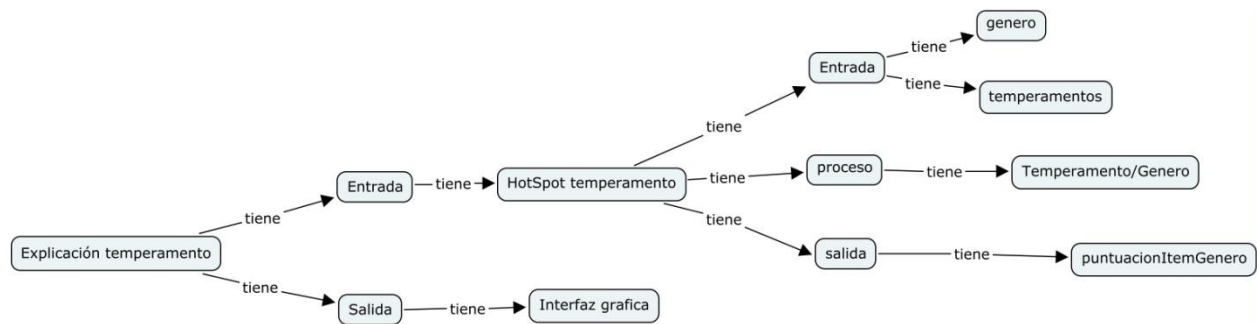


- Entradas, para poder generar la explicación necesitaremos los siguientes datos del punto caliente género: genero y temperamentos de la entrada, del proceso guardaremos todos los datos referentes a la similitud que existe entre un determinado genero del usuario y el género de una película, el género de un usuario y para finalizar también nos quedaremos con la puntuación final que le da la estrategia a la película.
- Salida, será una interfaz grafica en la que se nos muestre principalmente para cada género cual es el porcentaje de género que tiene la película, el usuario y su similitud.

En cuanto al proceso no habrá cálculos intermedios, lo único que haremos será mostrar todos los datos recogidos a partir de la entrada en forma de una tabla en la que pueda verificar los datos.

El sexto modelo de conocimiento de la explicación que veremos será el que esté relacionado con el punto caliente del temperamento, en la Figura 5.26 tenemos el modelo de conocimiento de esta explicación, donde:

Figura 5.26 Modelo de conocimiento de la explicación para la estrategia basada en el temperamento

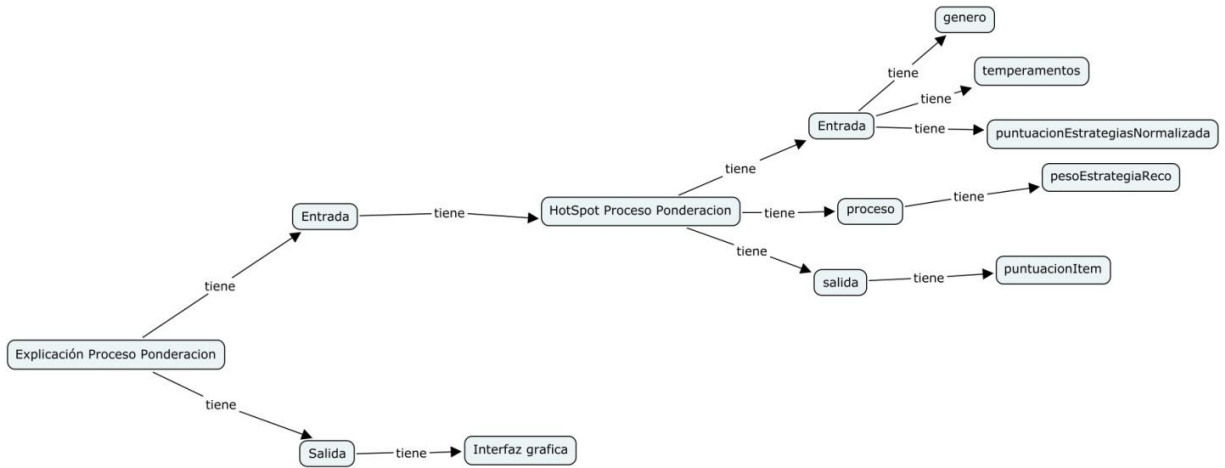


- Entradas, para poder generar la explicación necesitaremos los siguientes datos del punto caliente temperamento: genero y temperamentos de la entrada, del proceso guardaremos todos los datos referentes a la similitud que existe entre un determinado temperamento de la película y el temperamento del usuario, el temperamento de una película y para finalizar también nos quedaremos con la puntuación final que le da la estrategia a la película.
- Salida, será una interfaz grafica en la que se nos muestre principalmente para cada temperamento cual es el porcentaje de temperamento que tiene la película, el usuario y su similitud.

En cuanto al proceso no habrá cálculos intermedios, lo único que haremos será mostrar todos los datos recogidos a partir de la entrada en forma de una tabla en la que pueda verificar los datos.

El séptimo modelo de conocimiento de la explicación que veremos será el que esté relacionado con el punto caliente del proceso de ponderación, en la Figura 5.27 tenemos el modelo de conocimiento de esta explicación, donde:

Figura 5.27 Modelo de conocimiento de la explicación para el proceso de ponderación



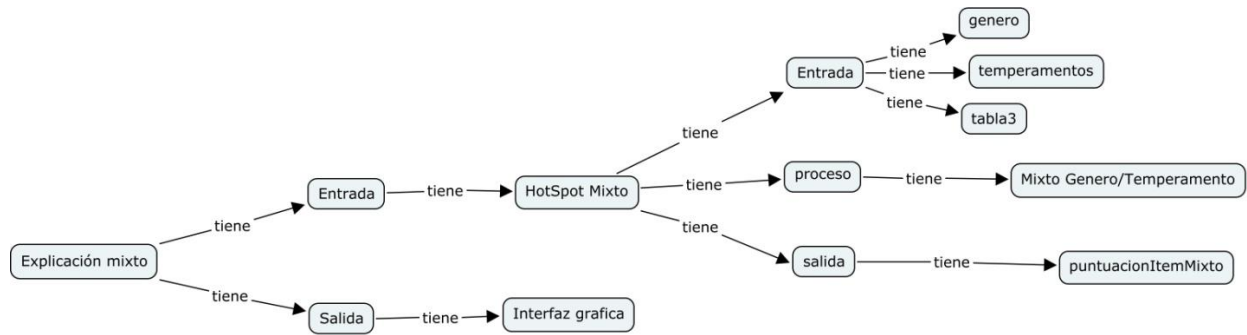
- Entradas, para poder generar la explicación necesitaremos los siguientes datos del punto caliente proceso de ponderación: genero, temperamentos de la entrada y puntuaciones normalizadas que ha obtenido la película en cada una de las estrategias, del proceso nos quedaremos con el valor referente al peso que le da a cada puntuación de las estrategias de recomendación y para finalizar también nos quedaremos con la puntuación final que le da el proceso a la película en función de cada estrategia.
- Salida, será una interfaz grafica en la que se nos muestre principalmente para puntuación de cada estrategia a la película, el peso de cada estrategia de recomendación y la influencia de cada estrategia en la puntuación final.

En cuanto al proceso a parte del proceso necesario para poder presentar los datos, necesitaremos calcular el valor de la influencia de cada estrategia en la recomendación, el cálculo de este valor es muy sencillo, usaremos la siguiente fórmula:

$$influencia(estrategia) = \frac{puntuacionItem(estrategia)}{\sum_{p=Mixta}^{Temperamento} puntuacionItem(p)}$$

Por último, el octavo modelo de conocimiento de la explicación que veremos será el que esté relacionado con el punto caliente mixto, en la Figura 5.28 tenemos el modelo de conocimiento de esta explicación, donde:

Figura 5.28 Modelo de conocimiento de la explicación para la estrategia mixta



- Entradas, para poder generar la explicación necesitaremos los siguientes datos del punto caliente género: genero, temperamentos de la entrada y los valores de la Tabla 5.4 de la sección 5.1, del proceso guardaremos para genero el valor de la combinación que existe entre los temperamentos del usuario y el valor de tabla3 y para finalizar también nos quedaremos con la puntuación final que le da la estrategia a la película.
- Salida, será una interfaz grafica en la que se nos muestra principalmente dos tipos de datos: una tabla en la que se verá cual es el valor de cada temperamento junto a la influencia que ha tenido ese valor en la puntuación del ítem y por otro lado para cada género de la película cual es su valor y cuál es la influencia de ese género en la puntuación final.

En cuanto al proceso habrá que realizar cálculos intermedios para poder determinar cuál es el valor de esas influencias, para la del género usaremos la siguiente fórmula:

$$influenciaGenero(genero) = \frac{mixtoGeneroTemperamento(genero)}{puntuacionItemMixto}$$

Para el valor de la influencia del temperamento usaremos la siguiente fórmula:

$$influenciaTempera(temperamento) = \frac{\sum_{g=Accion}^{Ciencia Ficción} pelicula(g) * tabla3(temperamento, g)}{puntuacionItemMixto}$$

Todavía quedarían dos modelos más, el proceso de conmutación y el referente a la estrategia sin estado de ánimo, pero ambos modelos de explicaciones son muy simples, tan solo usaran como entrada el estado de ánimo y generaran una explicación acorde.

Con todo esto ya tendríamos definidos nuestros modelos de explicaciones, el siguiente paso será junto a los modelos definidos anteriormente implementarlo en el lenguaje de programación que queramos, en este caso Java.

Capítulo 6 – Conclusiones y trabajo futuro

6.1 Conclusiones

En este proyecto fin de máster se ha hecho un estudio detallado de las explicaciones en los sistemas de recomendación y se ha presentado un modelo que permita la generación de explicaciones para los usuarios y para el ingeniero del conocimiento.

Se ha realizado un estudio del arte profundo de los tipos de explicaciones existentes en los sistemas inteligentes, no solo en los sistemas de recomendación, a su vez se ha realizado un estudio detallado del sistemas de recomendación para poder representar el conocimiento que usan y de esta forma poder generar un buen modelo para generar las explicaciones.

También se ha estudiado con detalle cuales son los aspectos que son más influyentes a la hora de generar las explicaciones, cuáles han de ser los aspectos en los que tiene que fijarse alguien que quiera generar explicaciones en un sistema inteligente.

Para finalizar se ha propuesto un modelo HotSpot Explanations, que le permitirá a un desarrollador generar explicaciones para su sistema de recomendación, podrá realizarlas de una manera sencilla siguiendo una serie de pautas de esta forma aumentara su productividad, además se aporta algo novedoso y es que permitirá a los ingenieros del conocimiento poder generar explicaciones que verifiquen el funcionamiento de sus sistemas.

6.2 Trabajo futuro

Como posibles líneas de trabajo futuro que continúen con este trabajo destacar:

- Desarrollar un lenguaje de especificación a partir de los diferentes modelos de conocimiento propuestos de forma que permitiera modelar la generación de las explicaciones de una forma formal. Desarrollo de una serie de herramientas que permitieran la edición y creación de los modelos y que además permitiera la validación y generación automática de código. De esta forma se reduciría bastante el esfuerzo que implica tener que añadir explicaciones a un sistema de recomendación.
- Evaluar la explicación generada para los usuarios finales en Moodvie, un aspecto importante de las explicaciones son el impacto que tienen en el recomendador y si de verdad cumplen con los objetivos para los que fueron diseñadas.

- Al estudiar un sistema de recomendación basado en los temperamentos, otra línea de investigación sería el estudio de explicaciones que se adaptaran dependiendo del temperamento del usuario.
- Mejorar el modelo propuesto para que permitiera generar explicaciones que no estén ligadas al proceso de recomendación, sino que pudieran usar otros tipos de explicación que fueran distintos al proceso de recomendación.

Referencias

- AAMODT, A. 2004. Knowledge-intensive case-based reasoning in CREEK. *Advances in Case-Based Reasoning, Proceedings*, 3155, 1-15.
- BALABANOVIC, M., SHOHAM, Y.(1997). Fab: content-based, collaborative recommendation, *Communications of the ACM* 40, no. 3, 66–72
- BASU, C., HIRSH, H., COHEN, W. (1998). Recommendation as classification: using social and content-based information in recommendation, in *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI'98)* (Madison, WI), American Association for Artificial Intelligence, pp. 714–720.
- BILGIC, M. and MOONEY, R. J. (2005). Explaining recommendations: Satisfaction vs. promotion. In *Proceedings of the Workshop Beyond Personalization*, in conjunction with the International Conference on Intelligent User Interfaces, pages 13–18.
- BILLSUS, D., PAZZANI, M.J. (1999). A Hybrid User Model for News Story Classification. In: *Proceedings of the 7th International Conference on User Modeling*, pp. 99-108
- BURKE, R. (2000) Knowledge-based recommender systems, *Encyclopedia of Library and Information Science* 69 no. 32, 180–200.
- BURKE, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370.
- CARBONERAS, D., ÁLVAREZ, C., MARTINEZ, M. (2012). MoodvieWeb
- CHANDRASEKARAN, B., TANNER, M. C. & JOSEPHSON, J. R. 1989. EXPLAINING CONTROL STRATEGIES IN PROBLEM-SOLVING. *Ieee Expert-Intelligent Systems & Their Applications*, 4, 9-&.
- CHEN, L., PU, P. (2002). Trust building in recommender agents. In *WPRSIUI* in conjunction with Intelligent User Interfaces, pages 93–100.
- CLANCEY, W. J. 1983. THE EPISTEMOLOGY OF A RULE-BASED EXPERT SYSTEM - A FRAMEWORK FOR EXPLANATION. *Artificial Intelligence*, 20, 215-251.

CRAMER, H., EVERS, V., RAMLAL, S., VAN SOMEREN, M., RUTLEDGE, L., STASH, N., AROYO, L. & WIELINGA, B. 2008. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction*, 18, 455-496.

CUNNINGHAM, P , DOYLE, D. & LOUGHREY, J. (2003). An Evaluation of the Usefulness of Case-Based Reasoning Explanation. In Ashley & Bridge (eds.) 122–130. Springer: Berlin Heidelberg.

DOYLE, D., TSYMBAL, A., and CUNNINGHAM, P. (2003). A review of explanation and explanation in case-based reasoning. Technical report, Department of Computer Science, Trinity College, Dublin.

FELFERNIG, A., GULA, B. (2006). Consumer behavior in the interaction with knowledge-based recommender applications. In ECAI 2006 Workshop on Recommender Systems, pages 37–41.

FELFERNIG, A., ISAK, K., SZABO, K., ZACHAR, P. (2007) The VITA financial services sales support environment , Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI'07), AAAI, 2007, pp. 1692–1699.

FIX, E., HODGES, J.L. (1951) Discriminatory analysis, nonparametric discrimination: Consistency properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas.

FRIEDRICH, G. & ZANKER, M. 2011. A Taxonomy for Generating Explanations in Recommender Systems. *Ai Magazine*, 32, 90-98.

GARRIDO, A., PEÑA, J. F., SALMERÓN, S. (2011). Moodvie: Recomendador individual para Facebook de películas basado en temperamentos y estados de ánimo

GREGOR, S. & BENBASAT, I. (1999). Explanations From Intelligent Systems: Theoretical Foundations and Implications for Practice. *MIS Quarterly* 23(4): 497–530.

HERLOCKER, J., KONSTAN, J., BORCHERS, A., RIEDL, J. (1999) "An Algorithmic Framework for Performing Collaborative Filtering". In Proceedings of the 1999 Conference on Research and Development in Information Retrieval (ACM SIGIR 99). Aug. 1999.

HERLOCKER, J. L., KONSTAN, J. A. & RIEDL, J. 2000. Explaining collaborative filtering recommendations. *CSCW 2000. ACM 2000 Conference on Computer Supported Cooperative Work*.

JANNACH, D. 2011. *Recommender systems: an introduction*, New York, Cambridge University Press.

KAMAL, A., WIJNAND, S.(2004) TiVo: making show recommendations using a distributed collaborative filtering architecture. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining

KEIRSEY, D. (2002). Por favor, comprendeme: Temperamento, carácter, inteligencia.

LACAVE, C. and DIÉZ, F. J. (2002). A review of explanation methods for bayesian networks. *The Knowledge Engineering Review*, 17:2:107–127.

LACAVE, C. and DIÉZ, F. J. (2004). A review of explanation methods for heuristic expert systems. *The Knowledge Engineering Review*, 17:2:133–146.

LINDEN, G., SMITH, B. and YORK, J. (2003) Amazon.com recommendations: item-to-item collaborative filtering, *Internet Computing*, IEEE 7, no. 1, 76–80.

MCSHERRY, D. (2001). Interactive Case-Based Reasoning in Sequential Diagnosis. *Applied Intelligence* 14: 65–76.

MCSHERRY, D. (2003) Similarity and compromise, Proceedings of the 5th International Conference on Case-Based Reasoning (ICCBR '03) (Trondheim, Norway).

MCSHERRY, D. (2004). Explaining the Pros and Cons of Conclusions in CBR. Funk, P. & González-Calero, P. (eds.) *Proceeding of ECCBR 2004* , 317–330. Springer: Berlin Heidelberg.

MCSHERRY, D. (2005). Explanation in Recommender Systems. *Artificial Intelligence Review* (This issue).

MAJCHRZAK, A. & GASSER, L. (1991). On using Artificial Intelligence to Integrate the Design of Organizational and Process Change in US Manufacturing. *AI and Society* 5: 321–338.

MASSIE, S, CRAW, S & WIRATUNGA, N (2004). Visualisation of Case-Based Reasoning for Explanation. In Gervás, P. & Gupta, K. M. (eds.) *Proceedings of the ECCBR 2004 Workshops* . Madrid. 135–144.

ONG, L. S., SHEPHERD, B., TONG, L. C., SEOWCHOEN, F., HO, Y. H., TANG, C. L., HO, Y. S. & TAN, K. 1997. The Colorectal Cancer Recurrence Support (CARES) System. *Artificial Intelligence in Medicine*, 11, 175-188.

PAZZANI, M., MURAMATSU, J., BILLSUS, D. (1996). Syskill & Webert: Identifying interesting web sites, *Proceedings of the 13th National Conference on Artificial Intelligence* (Portland, OR), 1996, pp. 54–61.

PAZZANI, M., BILLSUS, D. (1997). Learning and revising user profiles: The identification of interesting web sites, *Machine Learning* 27, no. 3, 313–331.

RICCI, F., ROKACH, L., SHAPIRA, B. & KANTOR, P. B. 2011. *Recommender Systems Handbook*. 1. ed. Boston, MA: Springer Science+Business Media, LLC.

RICHTER, M. M. (1995). The Knowledge Contained in Similarity Measures. Invited Talk at the First International Conference on Case-Based Reasoning, ICCBR’95, Sesimbra, Portugal.

ROTH-BERGHOFER, T. R. (2004). Explanations and Case-Based Reasoning: Foundational Issues, In Funk & González–Calero (eds.), 389–403. Springer: Berlin Heidelberg.

TINTAREV, N. 2010. Explaining recommendations. Tesis Doctoral.

SHARDANAND, U., MAES, P. (1995). Social information filtering: Algorithms for automating “word of mouth” , *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI ’95)*, vol. 1, 1995, pp. 210–217.

- SALTON, G, WONG, A and YANG C. S (1975), "A Vector Space Model for Automatic Indexing," *Communications of the ACM*, vol. 18, nr. 11, pages 613–620.
- SARWAR, B.M., KARYPIS, G., KONSTAN, J.A., and RIEDL, J. (2001) "Item-based collaborative filtering recommendation algorithms". In Proceedings of the 10th International World Wide Web Conference (WWW10), Hong Kong, May 2001.
- SINHA, R. and SWEARINGEN, K. (2002). The role of transparency in recommender systems. In Conference on Human Factors in Computing Systems, pages 830–831.
- SMYTH, B., COTTER, P. (2000) 'A Personalized TV Listings Service for the Digital TV Age'. *Knowledge-Based Systems* 13: 53-59.
- SORMO, F., CASSENS, J. & AAMODT, A. 2005. Explanation in case-based reasoning- perspectives and goals. *Artificial Intelligence Review*, 24, 109-143.
- SWARTOUT, W, 1983, XPLAIN: A system for creating and explaining expert consulting programs. *Artificial Intelligence* 21, 285–325.
- VIG, J., SEN, S. & RIEDL, J. 2009. Tagsplanations: explaining recommendations using tags. *Proceedings of the 14th international conference on Intelligent user interfaces*. Sanibel Island, Florida, USA: ACM.
- WICK, M. R. & THOMPSON, W. B. 1992. RECONSTRUCTIVE EXPERT SYSTEM EXPLANATION. *Artificial Intelligence*, 54, 33-70.
- ZANKER, M. & NINAUS, D. 2010. Knowledgeable Explanations for Recommender Systems. *Proceedings 2010 IEEE/ACM International Conference on Web Intelligence-Intelligent Agent Technology (WI-IAT)*.